

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE AUTOMATIZACIÓN
VIRTUAL PARA UNA MÁQUINA DE ESTAMPADO TEXTIL USANDO LA
ESTRATEGIA DE UN SOFTWARE IN THE LOOP.**

AUTOR

CRISTHIAN CAMILO TORRES GUEVARA

**UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD INGENIERÍA MECATRÓNICA
BOGOTA D.C.**

2016

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE AUTOMATIZACIÓN
VIRTUAL PARA UNA MÁQUINA DE ESTAMPADO TEXTIL USANDO LA
ESTRATEGIA DE UN SOFTWARE IN THE LOOP.**

AUTOR

CRISTHIAN CAMILO TORRES GUEVARA

**MONOGRAFÍA DE TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO MECATRÓNICO**

**UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD INGENIERÍA MECATRÓNICA**

BOGOTÁ D.C.

2016

DEDICATORIA

Dedico este proyecto principalmente a Dios y a la Virgen María por darme fuerzas, sabiduría, voluntad y perseverancia para afrontar las dificultades que se fueron presentando en el transcurso del proyecto.

Dedico este Proyecto de grado a mis padres y hermana que aunque algunos de ellos no pudieron estar en el proceso siempre fueron el motivo para seguir adelante y nunca desfallecer, por creer que puedo llegar hacer lo imposible posible y depositar su confianza en quien podría conseguir un nuevo orgullo para ellos.

AGRADECIMIENTOS

Agradezco a Dios primero que todo por darme la fortaleza y sabiduría para emprender este proyecto, a mi familia por su apoyo incondicional en el transcurso del desarrollo de todo el proyecto, a mis amigos y compañeros que me aportaron para el desarrollo del mismo.

Agradezco al Ing. Carlos Hector Gonzalez Villarraga, el Ing. Nestor Fernando Penagos, el Ing. Adonay varela, el Ing. Ruben hernandez y el Ing. David Herrera quienes fueron parte fundamental de este con sus aportes incondicionales.

CONTENIDO

| | pág. |
|--|----------|
| 1. ASPECTOS GENERALES | 1 |
| 1.1. INTRODUCCIÓN | 1 |
| 1.2. DESCRIPCIÓN DEL PROBLEMA | 2 |
| 1.3. FORMULACIÓN DEL PROBLEMA | 3 |
| 1.4. JUSTIFICACIÓN | 3 |
| 1.5. ESTADO DEL ARTE | 3 |
| 1.6. OBJETIVOS | 5 |
| 1.6.1. OBJETIVO GENERAL | 5 |
| 1.6.2. OBJETIVOS ESPECÍFICOS | 5 |
| 1.7. ALCANCES Y LIMITACIONES | 6 |
| 1.8. LÍNEA DE INVESTIGACIÓN DE PROGRAMA | 6 |
| 1.9. MARCO TEÓRICO | 7 |
| 1.9.1. ESTAMPACIÓN TEXTIL | 7 |
| 1.9.2. SOFTWARE IN THE LOOP | 8 |
| 1.9.3. 3Ds MAX® | 9 |
| 1.9.4. UNITY 3D® | 11 |
| 1.9.5. LAB VIEW® | 11 |
| 2. AUTOMATIZACIÓN DE LA MÁQUINA | 13 |
| 2.1. NECESIDADES DEL CLIENTE | 14 |
| 2.2. PROPUESTA DE AUTOMATIZACIÓN BRAZOS AXIALES | 15 |
| 2.2.1. CÁLCULOS PARA LA DETERMINACIÓN DEL ACTUADOR NEUMÁTICO ... | 16 |
| 2.2.2. MODELAMIENTO MATEMÁTICO DEL ACTUADOR NEUMÁTICO | 22 |
| 2.2.2.1. DESCRIPCIÓN DE LA PRUEBA | 24 |
| 2.2.3. SELECCIÓN DE COMPONENTE | 28 |
| 2.2.4. ELECTROVÁLVULA | 29 |
| 2.3. PROPUESTA DE AUTOMATIZACIÓN MESAS ROTACIONALES | 30 |

| | | |
|--------|--|----|
| 2.3.1. | CÁLCULOS MATEMÁTICOS PARA DETERMINAR EL MOTOR | 32 |
| 2.3.2. | MODELAMIENTO MATEMÁTICO DEL MOTOR ELÉCTRICO | 35 |
| 2.4.2 | ACTUADOR PARA MOVIMIENTO ROTACIONAL DE LAS MESAS | 38 |
| 2.4.3. | VARIADOR DE FRECUENCIA | 38 |
| 2.5. | DETERMINACIÓN DE LA RED NEUMÁTICA | 39 |
| 2.5.1. | COMPRESOR PARA LA RED NEUMÁTICA | 39 |
| 2.5.2. | CALCULO CAPACIDAD DE UN ACUMULADOR | 40 |
| 3. | VIRTUALIZACIÓN DE LA MÁQUINA EN UNITY | 42 |
| 3.1. | MODELADO DE LA MÁQUINA..... | 42 |
| 3.2 | . IMPORTACIÓN A UNITY®. | 45 |
| 3.4.1 | MOVIMIENTO ROTACIONAL..... | 47 |
| 3.3 | INTERFACES DE ENTRADA A UNITY®..... | 47 |
| 3.3.1. | PISTÓN..... | 48 |
| 3.3.2. | MOTOR..... | 49 |
| 3.5 | PROGRAMACIÓN DE MOVIMIENTOS | 50 |
| 3.5.1 | MOVIMIENTO VERTICAL | 51 |
| 4. | COMUNICACIÓN LABVIEW® CON UNITY® | 52 |
| 4.1. | COMUNICACIÓN UDP LABVIEW® A UNITY®..... | 52 |
| 4.2. | COMUNICACIÓN UDP UNITY® A LABVIEW®..... | 55 |
| 4.3. | APLICACIÓN DEMOSTRATIVA DE LA MÁQUINA | 57 |
| 5. | CONCLUSIONES..... | 62 |
| 6. | RECOMENDACIONES | 63 |
| | BIBLIOGRAFÍA..... | 64 |
| | ANEXOS..... | 67 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1. Técnicas encontradas comercialmente..... | 15 |
| Tabla 2. Propiedades de uno de los brazos. | 17 |
| Tabla 3. Propiedades de uno de los bastagos | 18 |
| Tabla 4. Propiedades del soporte que une las piezas. | 18 |
| Tabla 5. Comparación de actuadores..... | 29 |
| Tabla 6. Propiedades de las mesas..... | 32 |
| Tabla 7. Comparación de motores..... | 37 |
| Tabla 8. Tipos de compresores..... | 40 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1.Modelo de software in the loop | 9 |
| Figura 2.proceso de estampación textil | 13 |
| Figura 3.Automatización de brazos | 16 |
| Figura 4.Prueba de resistencia de uno de los brazos | 21 |
| Figura 5.Prueba de resistencia al soporte que recibe toda la carga | 21 |
| Figura 6.Laboratorio de obtención de modelos matemáticos | 23 |
| Figura 7.Descripción de la prueba..... | 25 |
| Figura 8.Obtención de los modelos | 26 |
| Figura 9.Transformada Z | 27 |
| Figura 10.Electroválvula 3/2 | 30 |
| Figura 11.Automatización de mesas..... | 31 |
| Figura 12.Mesas..... | 35 |
| Figura 13.Variador de frecuencia | 39 |
| Figura 14.Modelo de red neumática..... | 41 |
| Figura 15.Modelado de la máquina en SolidWorks | 42 |
| Figura 16.Modelado de máquina en 3Ds max..... | 44 |
| Figura 17.Principio de los pivotes..... | 45 |
| Figura 18.Importación de Assent | 46 |
| Figura 19.Assent visualizado en la escena principal..... | 46 |
| Figura 20. Movimineto rotacional..... | 47 |
| Figura 21.Interfaz de comunicación del pistón | 49 |
| Figura 22. Interfaz de comunicación del motor..... | 50 |
| Figura 23.Movimiento vertical | 51 |
| Figura 24.Prueba piloto de la comunicación | 52 |
| Figura 25.Comunicación | 54 |
| Figura 26.Comunicación retorno..... | 56 |
| Figura 27.Modelo de comunicación | 57 |
| Figura 28.Simulink pistón..... | 57 |
| Figura 29.Simulink motor | 58 |
| Figura 30.Comportamiento del pid pistón | 59 |
| Figura 31.Comportamiento del pid motor..... | 59 |
| Figura 32.Fase labview | 60 |
| Figura 33.Aplicación demostrativa | 61 |

LISTA DE ANEXOS

| | |
|--|----|
| Anexo 1. Manual se usuario electroválvula. | 67 |
| Anexo 2. Manual se usuario electroválvula | 68 |
| Anexo 3. Manual se usuario variador. | 69 |
| Anexo 4. Manual se usuario variador | 70 |
| Anexo 5. Ensamblaje total. | 71 |
| Anexo 6. Ensamblaje total 1 con detalle | 72 |
| Anexo 7. Ensamblaje total 2 con detalle. | 73 |
| Anexo 8. Ensamblaje total descriptivo | 74 |
| Anexo 9.Codigo C# recepción, modelos y movimientos de la máquina. | 75 |
| Anexo 10. Codigo C# envio de posición de los elementos virtuales | 84 |
| Anexo 11. Identificación del modelo de un motor. | 91 |

RESUMEN

Esta propuesta nace de la necesidad de mejorar los procesos manufactureros de una empresa textil colombiana, con la idea de implementar una simulación de la automatización de una máquina de estampación creando un laboratorio de control mediante un motor de juegos y un programa de visualización que permitirá el desarrollo de habilidades lógicas sobre el usuario.

Para la creación de la máquina de estampación virtual recurrimos a tres componentes esenciales: el diseño, el modelamiento y la programación. Todo esto en busca de un entorno virtual realista; teniendo en cuenta que la máquina física mecánica está compuesta por seis brazos y sus seis mesas, los brazos permiten un desplazamiento vertical donde se encuentran los marcos con el diseño y tinta de cada color para el proceso de estampado, las mesas se mueven de forma rotacional sobre el eje de la máquina, lo que permitirá estampar varias unidades en pocos minutos.

Este proyecto busca suplir la necesidad de mejorar tiempos de ejecución y calidad en el proceso de estampación, donde el operador de la máquina mecánica tiene como función bajar el brazo con el marco sobre la mesa que contiene la tela, además realiza un barrido con un escobillón (paleta de barrido), y posterior da un giro rotacional para cambiar de mesa e iniciar un nuevo estampado.

SUMMARY

The mechanical machine is composed of six arms and six tables, the arms allow a vertical displacement, there will be the frames with the design and color stamping, the tables move rotationally on the main axis of the machine according to the user's criteria. This will allow several printing units in a few minutes.

The operator of the machine has the function of lowering the arm with the frame on the table, It contains the fabric. It also makes a sweep with a broom (paddle sweep). At the same time, it allows the frame printing design on the fabric, it process is a repetition because each table must get down an arm.

This proposal was born from the necessity to improve the manufacturing processes of a Colombian textile company with the idea of implementing a simulation of automation. It permits to have a control laboratory on machine using a game engine and the visualization it will allow the development of logic user's skills.

1. ASPECTOS GENERALES

1.1. INTRODUCCIÓN

Este documento presenta una propuesta de automatización para una máquina de estampado textil, con el objetivo de establecer comunicación entre dos plataformas por medio de la estrategia de software in the loop, una de ellas será la encargada de controlar la máquina virtual y la otra su respectiva simulación. Este proyecto se lleva a cabo con el fin de proponer una estrategia buscando utilizar plataformas gráficas.

También responde a la propuesta de: automatización, correspondiendo fielmente al concepto de software in the loop que entrega un control sobre algún objeto o plataforma virtual, que para nuestro caso es la máquina de estampado textil; adicionalmente se establecen parámetros que pueden ser utilizados para la creación de un laboratorio virtual donde el usuario pueda evaluar las ventajas en la automatización de procesos.

A partir de la experiencia adquirida sobre los procesos de estampación textil y funcionamiento de la máquina de estampado se obtendrá un criterio de selección en controladores y actuadores para que pueda ser utilizada en una futura implementación.

El funcionamiento de la máquina se basa en la operación de la sujeción del marco contenedor de la imagen contra la prenda a procesar, la impresión gráfica es dada con la acción del operario sobre el brazo mecánico en sentido vertical, posteriormente el operario dará un giro rotacional a la mesa para continuar con la siguiente prenda. El proyecto busca convertirse en una herramienta útil al servicio de estudiantes y profesores, para analizar su respuesta frente a estrategias lógicas dentro del problema de automatización de las operaciones repetitivas anteriormente descritas.

En el capítulo inicial se verán los aspectos generales del proyecto, los alcances, sus limitaciones y una descripción detallada del proceso mecánico de estampación. Luego encontrarán las necesidades del cliente, la selección de los componentes a utilizar y sus respectivos cálculos mecánicos. Posteriormente se verán las interfaces de interacción, modelamiento y programación para la correcta virtualización dentro del entorno gráfico. Y finalmente se evidenciará la aplicación y la estrategia de comunicación del entorno gráfico y la plataforma de control.

1.2. DESCRIPCIÓN DEL PROBLEMA

La máquina de estampado mecánica es una de las herramientas más utilizadas en el mercado textil por las ventajas que puede ofrecer a los empresarios de este gremio, en la mayoría de casos en el mercado colombiano son usadas de forma mecánica debido a los altos costos que puede incurrir el obtener una automática o por el desconocimiento de estas, llevando a la empresa a invertir en mayor mano de obra y el proceso y sus resultados no sean tan eficientes.

La automatización es una solución hacia el desgaste humano de los procesos repetitivos que incurren para llevar algunos procesos en este caso de estampado, de esta forma provocaría el beneficio de empresarios que no podrían tener la facilidad de invertir en la compra de una maquina automatizada sin antes tener una proyección financiera para invertir en la automatización de uno de sus procesos industriales.

La manera más veraz y efectiva para comprobar que su inversión será de gran utilidad y no se convertirá en una mala compra, es mostrar por medio de una vista previa virtual el proceso automatizado que se suplirá, donde el usuario podrá hacer mover la máquina y sus componentes de forma que le genere satisfacción en la compra.

Para ello experimentaremos con la estrategia de software in the loop, la cual nos permite generar comunicación entre interfaces gráficas (Visual y control), las cuales producirían la interacción de la máquina de estampado de forma controlada y según las competencias del usuario; pero como esta es una creación reciente e innovadora aún tiene varias incógnitas a explorar, lo que nos lleva a la búsqueda y experimentación de estrategias para su realización.

1.3. FORMULACIÓN DEL PROBLEMA

¿Cómo se puede implementar una automatización virtual de una máquina de estampado textil mediante una estrategia de un software in the loop?

1.4. JUSTIFICACIÓN

Generar la comunicación entre dos plataformas una de ellas visual y la otra de control es crear una nueva estrategia para la visualización de un proceso automatizado, logrando así abrir un campo innovador de investigación dentro del programa de ingeniería mecatrónica de la Universidad Piloto de Colombia, basándose en la estrategia de software in the loop, generando la correcta comunicación entre la virtualización de una máquina de estampado textil y su plataforma de control, ofreciendo así un laboratorio para evaluar diversas competencias de automatización. Y quizá a largo plazo se pueda usar como herramienta de mercadeo.

1.5. ESTADO DEL ARTE

Con la llegada de la revolución industrial a nuestro mundo en 1820 dejamos atrás la antigua economía de la agricultura, aldeas y artesanías, generando en consecuencia que las empresas produzcan en poco tiempo grandes cantidades, lo que ha reducido el desgaste humano por realizar tareas manufactureras de forma repetitiva, así que al revisar las funciones que compromete un esfuerzo físico del empleado, se propone una simulación en donde se supla este desgastes [2], por medio de la realización de un entorno de ayuda para la simulación en la realidad

¹ [2] KESAVADAS, T., 1995. *Virtual Reality Based Interactive Automation of a Robotic Die Grinding Operation*. Order No. 9531955 ed. Ann Arbor: The Pennsylvania State University.

² [17] MCLELLAN, Hilary. *Virtual realities. Handbook of research for educational communications and technology*, 1996, p. 457-487.

³ [7] CARRILLO, M.D., 2013. *VizLab: The Design and Implementation of an Immersive Virtual Environment System using Game Engine Technology and Open Source Software*. Order No. 1539751 ed. Ann Arbor: The University of Texas - Pan American ProQuest Dissertations & Theses A&I. ISBN 9781303157882.

virtual, que es creada por el ordenador donde el usuario pueda evaluar que tan viable será el proyecto y qué defectos podemos arreglar en post producción.

La realidad virtual es un conjunto de tecnologías electrónicas emergentes, con aplicaciones en una amplia gama de campos. [17] Convirtiéndose en una excelente herramienta, ya que nos permite realizar una previa visualización de los productos antes de invertir factores como tiempo y dinero en su construcción. Logrando hacer un previo laboratorio donde nos muestre las fallas y los aciertos de los diseños que vayamos a plantear.

En las industrias, la automatización virtual es más que la prueba principal de funcionamiento de cualquier tipo de proyecto, este se verá comprometido a ser la plataforma base para la creación física de la máquina. La automatización virtual en muchos casos tiene el objetivo de convertirse en un laboratorio virtual de apoyo en la investigación disminuyendo gastos y esfuerzos [7], logrando que controladores y hardware pueden ser probados en un entorno simulado antes de la producción para asegurar componentes y sistemas de fidelidad.[9]

Una de las estrategias más reconocidas en la automatización es Software In The Loop, pues es una de las mejores posicionadas en el mercado por su facilidad de uso, debido que permitir la comunicación entre entornos para verificar un sistema mediante una simulación sin comprometer de forma económica el proyecto que se esté realizando [3].

El motor de Juego es una aplicación que nos permite generar movimientos a la simulación realista usando lenguajes de programación orientados a la estrategia de comunicación, con la estipulación previa de los modelos matemáticos en los actuadores o elementos que componen tal proyecto. Se conoce que es una herramienta multiproceso con un sistema orientado al juego [6], además se usa en posibles actualizaciones del entorno permitiendo implementar nuevos usos y características para la posible evaluación de competencias educativas. La plataforma que permitirá diseñar estrategias de automatización es escogida con el objeto de interactuar en un entorno de programación gráfico, llevando al usuario a contemplar ámbitos y experiencias educativas nuevas, a partir de un programa eficaz que comprometerá directamente la automatización a la interpretación encontrada por el usuario y la necesidad requerida.

La estrategia de software in the loop es aprovechada en una serie de pruebas, un ejemplo muy particular es la validación de un controlador supervisor por un conector en paralelo a través del camino de un vehículo híbrido eléctrico, evidenciando que por medio del software se demuestra que funciona de forma correcta para la reducción de gases con resultados que satisfacen todas los requisitos de seguridad y funcionales para reducir el consumo de combustible.[3]

Otra prueba encontrada es el diseño de un simulador para obtener pruebas de rendimiento de un vehículo y dar formación a los conductores logrando que los resultados de los experimentos muestren la eficacia y el rendimiento en tiempo real del simulador que puede realizar un "conductor del vehículo de carretera" en simulación cerrada. [5]

1.6. OBJETIVOS

1.6.1. OBJETIVO GENERAL

Diseñar e implementar un sistema de automatización virtual para una máquina de estampado textil usando la estrategia de un software in the loop.

1.6.2. OBJETIVOS ESPECÍFICOS

- Determinar y analizar los elementos que permiten la automatización mecánica y electrónica de la máquina.
- Obtener los modelos matemáticos de los elementos principales de la máquina los cuales deben ser implementados en la interfaz simulación.
- Implementar una aplicación que permita al usuario la interacción con la máquina estampadora virtual para procesos de automatización usando software in the loop.

⁴ [9]2014. GM Global Technology Operations LLC; Patent Issued for Vehicle Simulation System with Software-in-the-Loop Bypass Control. *Computers, Networks & Communications*, Jan 02, pp. 2938 ProQuest Career and Technical Education; ProQuest Computing; ProQuest Science Journals.

⁵ [3]LOR, L.A., 2013. *Supervisory Controller Validation for a Plug-in Parallel-through-the-Road Hybrid Electric Vehicle by Software-in-the-Loop Testing*. Order No. 1543709 ed. Ann Arbor: Wayne State University ISBN 9781303312649.

⁶ [6]CARR, D.A., 2009. *Towards Automatic Parallel Game Engine Architectures*. Order No. 1464428 ed. Ann Arbor: University of Nevada, Reno ProQuest Dissertations & Theses A&I. ISBN 9781109145281.

¹⁰ [5]KANG, Y., et al, 2013. Design of Driver-in-the-Loop Simulator for Off-Highway Truck. *Sensors & Transducers*, 12, vol. 161, no. 12, pp. 401-407 ProQuest Computing; ProQuest Science Journals; ProQuest Telecommunications. ISSN 23068515.

- Generar los conectores entre la máquina estampadora textil virtual y el software LabView que permita establecer las aplicaciones de automatización.
- Implementar una automatización demostrativa de la máquina estampadora textil.

1.7. ALCANCES Y LIMITACIONES

- Definen el cubrimiento del proyecto y los compromisos asumidos por el estudiante quien utilizara un software de modelado, animación y renderización en 3D para realizar la visualización y simulación del comportamiento de la máquina.
- Se usará un software motor de juego para la implementación de los modelos matemáticos debido a que permite la implementación de ecuaciones bajo lenguajes de programación y así tener un acercamiento realista de los movimientos de la máquina, adicionalmente incursionar en el uso de nuevas plataformas nunca antes utilizadas.
- Se busca implementar el software LabView® para generar la apertura a un nuevo campo de automatización, el cual no ha sido utilizado en el programa de ingeniería mecatrónica.

1.8. LÍNEA DE INVESTIGACIÓN DE PROGRAMA

- Automatización y domótica.

Este proyecto busca innovar en la estrategia de Software In The Loop, que nos permitirá realizar el control sobre una automatización virtual, que en este caso será netamente grafica buscando satisfacer las necesidades iniciales del cliente.

La mayor importancia para este proyecto es el uso y exploración de nuevas herramientas que puedan incurrir en un aporte social y educativo, comprobando que es posible la creación de un entorno grafico que pueda llegar a evaluar competencias de automatización.

1.9. MARCO TEÓRICO

1.9.1 ESTAMPACIÓN TEXTIL

La estampación textil inició hace siglos con el fin de tener una distinción entre las diferentes tribus o sociedades; este proceso es una de las invenciones más importantes y versátiles para implementar el diseño, el color y diversos tipos de telas. Esta técnica es la combinación de arte, ingeniería, tecnología y creatividad e imaginación para los diseños; donde con estas características se implementó en las telas de vestir y actualmente en el mercado se utilizan en tapicería, productos para el hogar, entre otros.

El primer proceso de estampación que se desarrolló en las primeras civilizaciones fue el Batik, donde se estableció patrones únicos y muy similares al estampado textil; Los antiguos egipcios usaron diseños con gradados en los extremos de cilindros para estampar alrededor de los 400 DC. Al final del siglo XIV en Europa se obtuvo la primera evidencia del estampado textil; en 1440 la familia de los estampadores textiles (Tuchdruckers) fue muy conocida en Alemania; en Estados Unidos se practicó este proceso de estampación con bloques de madera en diferentes ciudades a mediados de 1770, después en 1783 un escocés realizó una gran creación y es el rodillo con un grabado. [22]

El sistema del rodillo grabado consta con el acabado de los tejidos o hilado donde se tiñe aplicando un colorante soluble, natural o sintético bajo varios parámetros en una o más zonas en el mismo material (Cotton Incorporated, 2003). Los colorantes reactivos son muy conocidos por su evolución técnica, esto se debe a las propiedades de solidez, la compatibilidad, la adaptación en diversos equipos aptos para tinturas y las técnicas de aplicación; estos colorantes se aplican más que todo en materiales de algodón, seda y viscosas, pero también se ha logrado en lino, modal, bambú, entre otras fibras.

1.9.2 SOFTWARE IN THE LOOP

Es una estrategia que permite la comunicación entre dos plataformas para implantar un control sobre la simulación precisa del modelo matemático a partir del sistema físico. Es de importancia conocer que la cantidad de tiempo del CPU utilizado por el proceso de simulación podría llegar a ser igual al utilizado por el desplegado o real, [10] su modelo se ve reflejado en la figura 1.

Propósito de software:

- inclusión de la funcionalidad algorítmica de un modelo para su correcto control.
- Aumentar la velocidad de simulación mediante la inclusión del código compilado en lugar de modelos interpretativos
- Generar la simulación que garantice que un modelo matemático funcione de forma idéntica que en un objeto real.
- Garantizar que un algoritmo en el entorno de modelado funcionará idénticamente a ese mismo algoritmo se ejecuta en un controlador de producción

¹² [10]COLLINS, D.E. and GEORGE, A.D., 2001. Parallel and Sequential Job Scheduling in Heterogeneous Clusters: A Simulation Study using Software in the Loop. *Simulation*, Nov, vol. 77, no. 5, pp. 169 ProQuest Science Journals. ISSN 00375497.

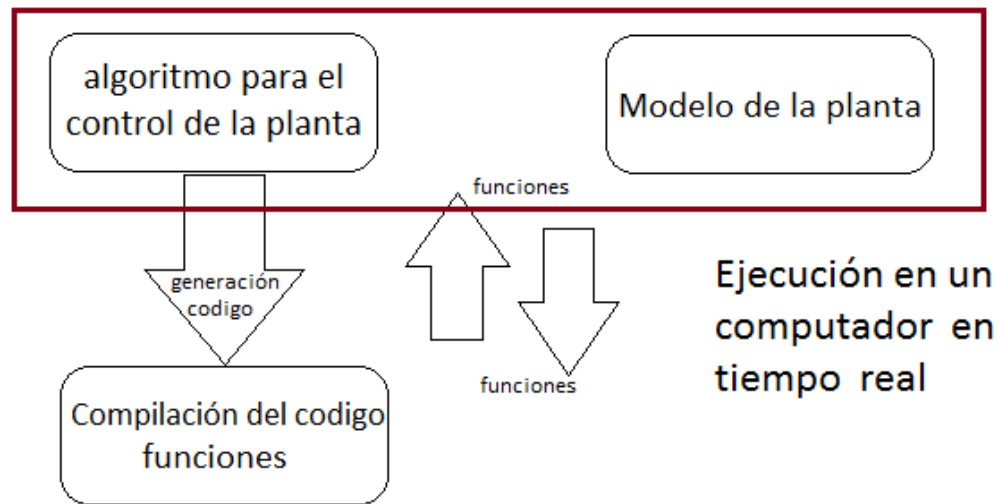


Figura 1, Modelo de software in the loop[1]

1.9.3 3Ds MAX®

Es un programa de creación de gráficos, modelado, simulación y animación en 3D desarrollado por Autodesk, su arquitectura es basada en plugins. Este programa posee 5 características principales que permiten una mejor manipulación del usuario [19]

Características:

Animación

- Mejoras de relleno
- Animación de relleno de multitudes
- Herramientas de animación y manipulación de personajes
- Herramientas de animación general

¹³ [1] RILEY, Patrick F.; RILEY, George F. Next generation modeling III-agents: Spades---a distributed agent simulation environment with software-in-the-loop execution. En Proceedings of the 35th conference on Winter simulation: driving innovation. Winter Simulation Conference, 2003. p. 817-825.

¹⁴ [19] REINHART, Christoph; BRETON, Pierre-Felix. Experimental validation of Autodesk® 3ds Max® Design 2009 and DAYSIM 3.0. Leukos, 2009, vol. 6, no 1, p. 7-35.

- Deformidades animados

Modelado y texturizado

- Compatibilidad con las nubes de puntos
- Sombreador shaderFX
- Herramientas de colocación
- Chaflán cuádruple
- Modelado de mallas y superficies
- Asignación y edición de texturas
- Sombreado y diseño de materiales

Renderización 3D

- Renderización mejorada de sombreado activo
- Rendimiento de ventana gráfica acelerada
- Cámara estereoscópica
- Opciones de renderización integradas
- Sistema render pass
- Núcleo de gráficos acelerado
- Editor de composición slate

Dinámicas y efectos

- Flujo de partículas
- MassFx
- Cabello y pelaje

Interfaz de usuario, flujo de trabajo y estructura productiva

- Gestión de escenas mejorada
 - Script python
 - Interfaz de usuario configurable
 - Igualación flexible de cámaras
 - Transferencia de datos e integración de estructuras de trabajo
 - Integración de composiciones
-

- Flujos de trabajo de colaboración con contenedores

1.9.4 UNITY 3D®

Es una plataforma de desarrollo flexible y poderoso para crear juegos y experiencias interactivas bajo el formato 3D y 2D (Multiplataforma). [18]

Una de las más importantes características de esta plataforma es que permite la programación de entornos y objetos bajo un lenguaje de programación convencional, facilitando la experiencia del usuario y sus posibles aplicaciones, también se puede establecer comunicación entre varios programas como lo es maya, blender, etc. [24]

Una de las mayores ventajas de UNITY 3D es poder proporcionar cambios a cualquier tipo de objeto creado con los programas anteriormente nombrados, donde se actualizarán automáticamente durante el proyecto sin necesidad de importarlo manualmente, incluso llega a exportar el proyecto con sus escenas a un ejecutable Windows o plataforma mac, Xbox, psp, Android entre otras.

Al ser un programa compatible con varias plataformas nos permite generar aplicaciones para cada una de ellas. Una posible aplicación es la comunicación entre un dispositivo móvil y un servidor provocando la interacción no presencial de un laboratorio virtual.

1.9.5 LAB VIEW®

LabView es una plataforma de programación gráfica que ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Ofrece integración sin precedentes con software legado existente, IP y hardware al aprovechar las últimas tecnologías de cómputo. [12] LabView ofrece

¹⁵ [18]CASIS, E., et al. Virtual automation. Clinical leadership & management review: the journal of CLMA, 2000, vol. 15, no 2, p. 89-91.

¹⁶ [24] Geovanny Méndez, Enmanuel Obviedo, Gabriel Fallas, Cristián Vega, Abel Méndez. Análisis de las herramientas Unity y Blender para el desarrollo de videojuegos con un enfoque educativo. Escuela de computación, tecnologico Costa rica. copyright, 2003.

herramientas para resolver los problemas de hoy en día y la capacidad para la futura innovación, más rápido y de manera más eficiente. [16]

LabView es uno de los lenguajes que se utilizan para la automatización de procesos, no sólo por su facilidad de programación, sino también porque presenta un ambiente similar a los dispositivos reales, pero que sólo están presentes dentro de un programa de cómputo. [19]

¹⁶ [12]KESAVADAS, T., 1995. *Virtual Reality Based Interactive Automation of a Robotic Die Grinding Operation*. Order No. 9531955 ed. Ann Arbor: The Pennsylvania State University ProQuest Dissertations & Theses A&I, <http://www.ni.com/labview/esa/>.

¹⁷ [16]NEUGEBAUER, J.-., 1992. Virtual Reality - More than just Simulation. *The Industrial Robot*, vol. 19, no. 3, pp. 30 ABI/INFORM Global; ProQuest Research Library; ProQuest Science Journals. ISSN 0143991X.

¹⁸ [19]REINHART, Christoph; BRETON, Pierre-Felix. Experimental validation of Autodesk® 3ds Max® Design 2009 and DAYSIM 3.0. *Leukos*, 2009, vol. 6, no 1, p. 7-35.

2. AUTOMATIZACIÓN DE LA MÁQUINA

En el proceso de estampado el operador de la máquina debe bajar el brazo con el marco sobre la mesa que contiene la tela, luego realiza un barrido con el escobillón o paleta de barrido para transferir la tinta sobre la tela, posteriormente realiza un giro rotacional para cambiar de mesa e iniciar el proceso nuevamente. Como se muestra en la figura 2.

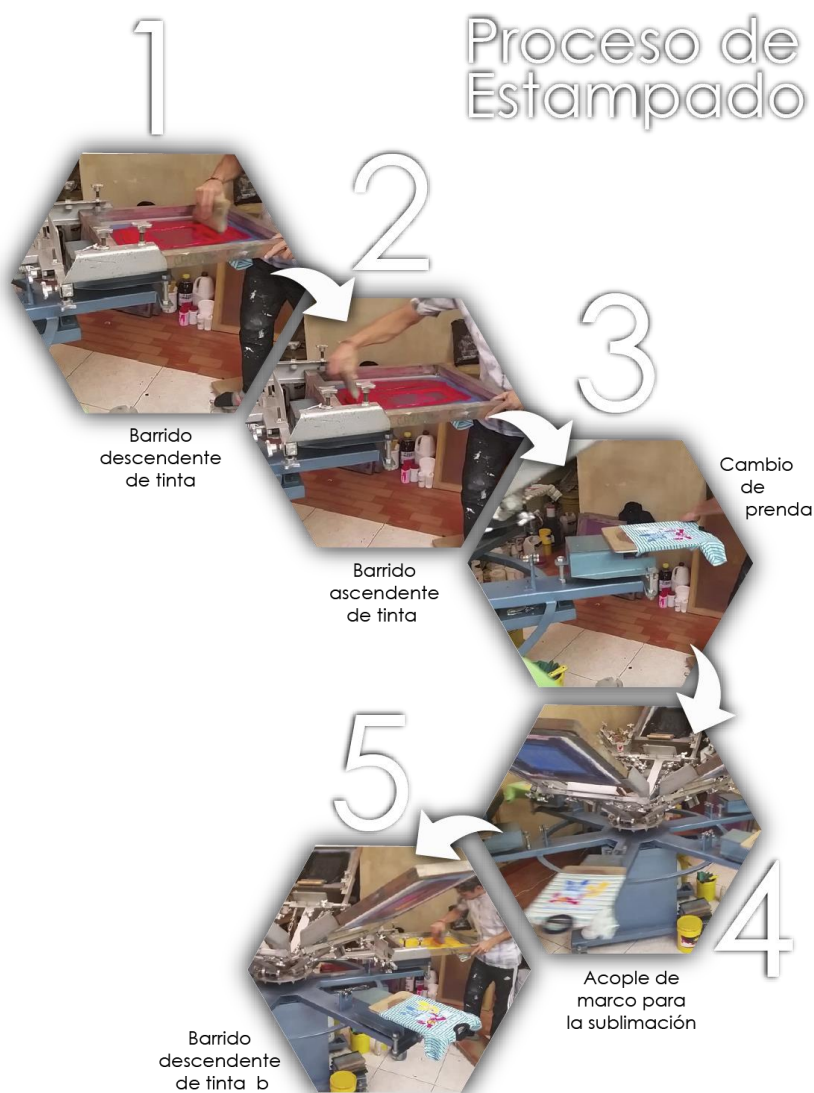


Figura 2, proceso de estampación textil

Se quiere automatizar el proceso de estampado mencionado anteriormente, ya que el operador realiza un trabajo mecánico y repetitivo; a excepción del barrido de la tinta debido a que es un proceso que por experiencia, una máquina no tiene la capacidad aún replicar de forma exacta las técnicas especiales de estampado.

2.1. NECESIDADES DEL CLIENTE

La mayoría de la producción industrial de un país en desarrollo, viene de las pequeñas y medianas empresas, más conocidas como las pymes, pero estas por ser tan novatas en un mercado tienen grandes monstruos llamados multinacionales que las obliga a que sus procesos y productos sean de mejor calidad, lo que las ha llevado a invertir en equipos y maquinarias que les permitan ser más competitivas en el mercado y tener un valor diferencial en este.

Para este trabajo se ha escogido una necesidad saliente de la industria textil acentuando el campo del estampado de prendas; específicamente la máquina mecánica que genera la sublimación de las prendas más conocida en el gremio como el pulpo. Donde encontramos que durante el proceso de estampado el operario debe realizar movimientos repetitivos que consiste en bajar y subir el brazo sobre las mesas y posteriormente realizar un movimiento rotacional de las mesas para cambiar de prenda.

Lo que busca este proyecto es generar movimientos autónomos y eficaces que faciliten el trabajo por medio de la automatización, generando una serie de ventajas como: la eliminación de trabajos rutinarios, la seguridad del operario, la mejora en la calidad del producto, un mayor rendimiento de la producción, la disminución de la cantidad de piezas defectuosas, la reducción del tiempo de manufactura y la creación de sistemas de información que permite evaluar el rendimiento de la empresa para implementar mejoras en sus procesos.

Una de las principales problemáticas por la cual las empresas de la industria textil no invierten en este tipo de máquinas es el elevado monto de la inversión, ya que se puede convertir de un capital a un gasto innecesario, lo que las lleva a realizar un estudio de la inversión y una proyección donde pueda visualizar el retorno de esta (ROI).

2.2. PROPUESTA DE AUTOMATIZACIÓN BRAZOS AXIALES

A base del trabajo de cotizaciones sobre fabricantes a nivel internacional y sus recomendaciones, se busca establecer la propuesta adecuada, eficaz y sugerida para la automatización de los brazos axiales como se muestra en la tabla 1, dejando escogida como la técnica de un pistón neumático por su rapidez en la interacción del movimiento y fácil implementación.

Tabla 1, técnicas encontradas comercialmente.

| | Elevación de los brazos | | |
|----------------------|-------------------------|------------------|------------------|
| | M&R | Sroque | Casa del screen |
| Técnica de elevación | Pistón neumático | Pistón neumático | Pistón neumático |

El modelo de automatización que se plantea para la máquina de estampado de 6 brazos, deberá contar con una base en la que se encuentran empotrados 5 brazos permitiendo así que en lugar del sexto brazo se encuentre la entrada y salida de las prendas a estampar, adicionalmente deberá contar con un acople de un pistón neumático, hacia el soporte que contiene la conexión mecánica de cada uno de los brazos como se muestra en la figura 3.

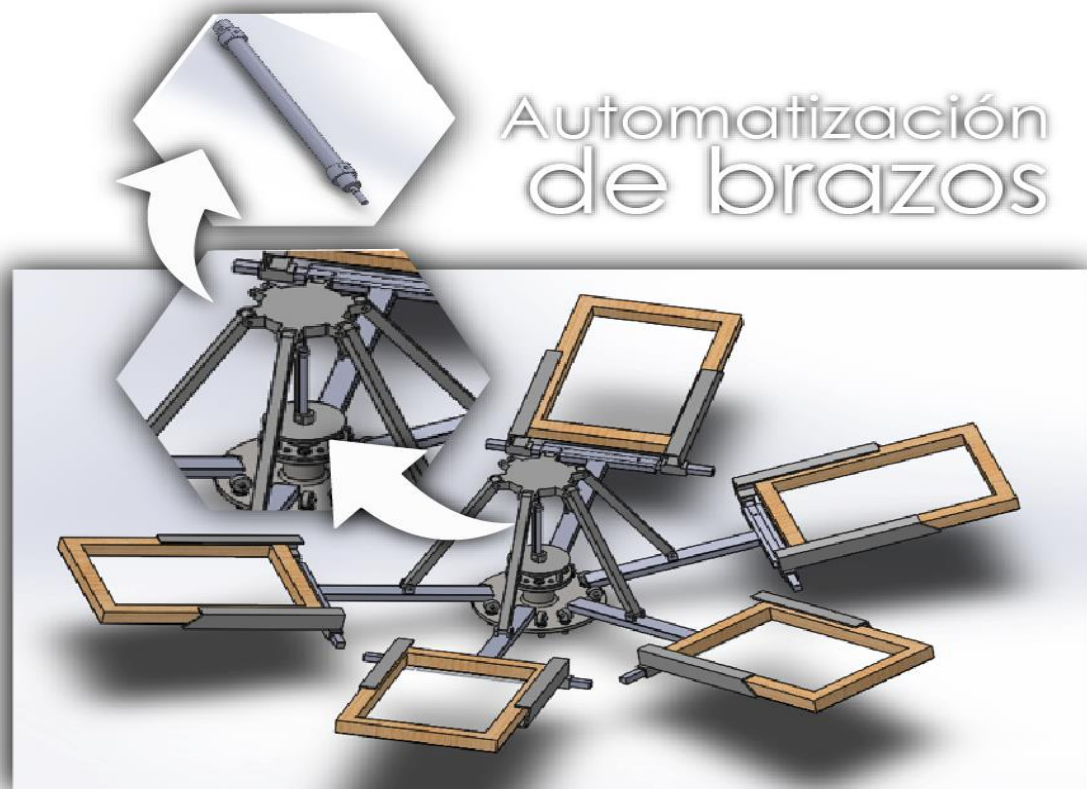


Figura 3, Automatización de brazos

El anterior modelo se plantea en base de los requerimientos de la empresa y de sus experiencias encontradas en la fabricación de prendas, donde se necesita remover únicamente el sexto brazo para no generar ningún sobre proceso en un estampado especial, ya que comúnmente se realizan estampados con un máximo de 5 tintas que proporcionalmente estarán ubicadas en los cinco restantes brazos.

2.2.1. CÁLCULOS PARA LA DETERMINACIÓN DEL ACTUADOR

NEUMÁTICO

Para ejercer una correcta aplicación de cálculos se implementó desde el programa Solidworks las propiedades de masa de los elementos involucrados en la automatización siendo los brazos, sus respectivos acoples a el eje principal y la

pieza que concentrará las masas hacia el pistón neumático, como se muestra en las Tablas 2 a 4.

Tabla 2, Propiedades de uno de los brazos.

| | Propiedades de un brazo | | |
|--|--------------------------|---------------------|----------------------|
| | X | Y | Z |
| Centro de masa | -385.89 mm | 872.13 mm | 222.79 mm |
| Momentos de inercia (g*mm) | $I_x=(0.87,-0.03,-0.50)$ | $I_y=(-0.5,0,0.87)$ | $I_z=(0.03,1,-0.01)$ |
| Momentos de inercia (g*mm) | $P_x=(1554250.44)$ | $P_y=(33138470.64)$ | $P_z=(34463971.55)$ |
| Masa: 1107.04g | | | |
| Volumen: 1107037.63 mm ³ | | | |
| Área de superficie 211145.83 mm ² | | | |

Tabla 4, Propiedades de unos de los bastagos.

| | Propiedades de soporte de los brazos | | |
|---|--------------------------------------|---------------------|---------------------|
| | X | Y | Z |
| Centro de masa | 239 mm | 1111.01 mm | 138.14 mm |
| Momentos de inercia (g*mm) | $I_x=(-0.5,0.82,-0.29)$ | $I_y=(-0.5,0,0.87)$ | $I_x=(0.71,0,0.41)$ |
| Momentos de inercia (g*mm) | $P_x=(69900.37)$ | $P_y=(10509411.83)$ | $P_z=(10509485.46)$ |
| Masa: 466.49g | | | |
| Volumen: 466493.36 mm ³ | | | |
| Área de superficie 65811.95 mm ² | | | |

Tabla 5, Propiedades del soporte que une las piezas.

| | Propiedades del soporte del actuador | | |
|--|--------------------------------------|--------------------|--------------------|
| | X | Y | Z |
| Centro de masa | 0 mm | 1310.38 mm | 0 mm |
| Momentos de inercia (g*mm) | $I_x=(1,0,-0.01)$ | $I_y=(-0.01,0,-1)$ | $I_x=(0,1,0)$ |
| Momentos de inercia (g*mm) | $P_x=(5223282.29)$ | $P_y=(5223286.37)$ | $P_z=(9975498.43)$ |
| Masa: 1590.16 g | | | |
| Volumen: 1590156.52 mm ³ | | | |
| Área de superficie 133377.49 mm ² | | | |

Para la determinación del pistón inicialmente se extraen las masas de todos los elementos comprometidos en la automatización de los brazos como se muestran en las ecuaciones (2.1 a 2.4):

Masas de las piezas

$$1.59kg \quad (2.1)$$

$$0.466 \times 5 = 2.33kg \quad (2.2)$$

$$1.11 \times 5 = 5.55 kg \quad (2.3)$$

$$total\ de\ las\ masas = 9.47 kg \quad (2.4)$$

Para este caso se buscó establecer inicialmente la masa total (2.4) que el actuador deberá mover, posterior se realiza la determinación de la aceleración para poder establecer la fuerza (masa por aceleración).

Se conoce que la aceleración se ve expresada en función de la posición sobre un delta de tiempo como se muestra en la ecuación (2.6), al ser despejada de la ecuación (2.5), entregando como resultado la aceleración que tiene los cuerpos (2.7).

Determinación de la aceleración

$$x = V_0 * t + \frac{1}{2} at^2 \quad (2.5)$$

$$a = \frac{x}{\frac{1}{2} * t^2} \quad (2.6)$$

$$a = 0,25 m/s^2 \quad (2.7)$$

La determinación de la fuerza se ve expresada en función de la masa por la aceleración del cuerpo (2.8), una vez encontrada (2.9) se le aplica como factor de seguridad 1.2 como se muestra en la ecuación (2.10) y (2.11).

Determinación de la fuerza

$$F = m * a \quad (2.8)$$

$$F = 2,5675KN \quad (2.9)$$

$$\text{factor de seguridad } 1.2 \quad (2.10)$$

$$F_{final} = 3,018KN \quad (2.11)$$

Una vez encontrada la fuerza se despeja de la ecuación 12 el área para su posterior discriminación en función del radio como se muestra en la ecuación (2.15). Más tarde se supone la presión de trabajo seleccionando la presión industrial comercial, encontrando como resultado final el radio del pistón neumático a utilizar como se muestra en la ecuación (2.16)

Determinación del diámetro del pistón

$$P = \frac{F}{A} \quad (2.12)$$

$$A = \frac{F}{P} \quad (2.13)$$

$$\pi r^2 = \frac{F}{P} \quad (2.14)$$

$$r = \sqrt{\frac{F}{P * \pi}} \quad (2.15)$$

se supone una presión de trabajo de 6 bares siendo la presión industrial más comercial

$$r = 12mm \quad (2.16)$$

Bajo el conocimiento de la función de presión (fuerza / área) se establece el radio del actuador neumático a utilizar dejando como resultado un radio de 12 milímetros.

Adicionalmente se practicó una prueba de deformación a los elementos más susceptibles a fallar por medio del programa Soldworks® sin importar que dentro de los objetivos no está la fabricación real de la máquina, como se muestran en las figuras 4 y 5, se confirmó que era 100% funcional y podría estar en uso sin estar comprometidos mecánicamente en sus propiedades de resistencia de materiales.

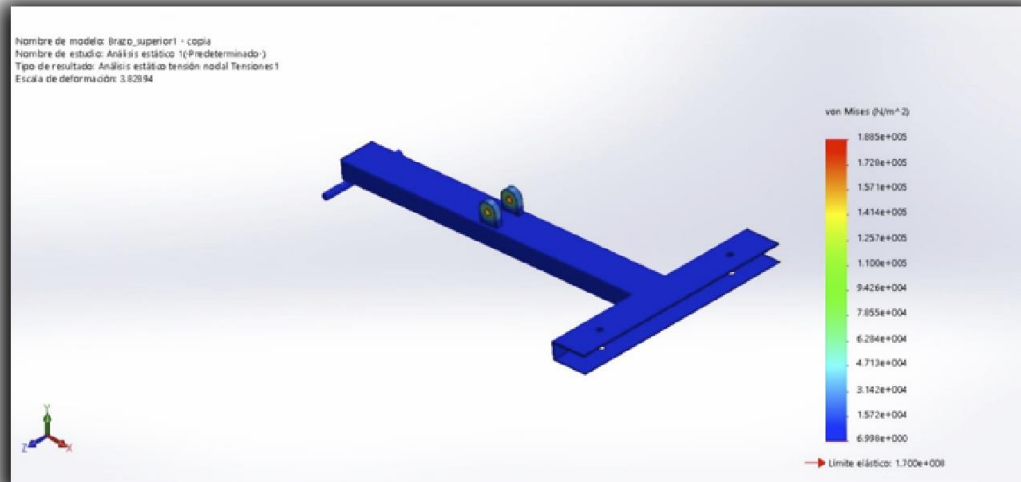


Figura 4, Prueba de Resistencia a uno de los brazos.

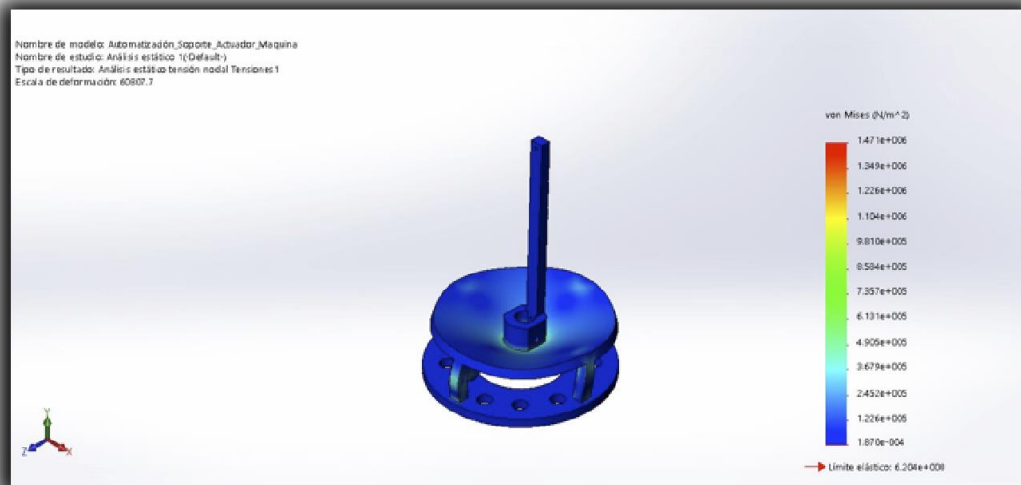


Figura 5, Prueba de Resistencia al soporte que recibe toda la carga.

2.2.2. MODELAMIENTO MATEMÁTICO DEL ACTUADOR NEUMÁTICO

Para establecer de forma correcta el modelo matemático fue necesario implementar un laboratorio de pruebas, debido a que teóricamente ningún modelo se ajustaba a las necesidades según el entorno visual; para ello se realizaron 8 pruebas a fallo y error obteniendo un aproximado de su ecuación característica que modelan su movimiento. Es importante aclarar que dicho laboratorio fue una escala del proceso original, debido a que físicamente se contó únicamente con las herramientas suministradas por la universidad, como se muestra en la figura 6.

Aspectos técnicos para la prueba:

- Actuador Festo (diámetro 8mm y recorrido 15 cm)
- Presión de 6 bares
- Carga de 3.15 kg
- Sensor ultrasónico Ha 044H
- Arduino UNO
- Válvulas neumáticas Festo
- Manómetro Festo
- Báscula
- Calibrador / pie de rey

Laboratorio para obtención de Modelos Matemáticos



Figura 6, Laboratorio para obtención de modelos matemáticos.

2.2.2.1. DESCRIPCIÓN DE LA PRUEBA

Esta prueba está compuesta por 2 fases la primera es el ensamblaje adecuado del mecanismo que asegura una obtención adecuada de datos y la segunda donde se toma datos de cada una de las interacciones que puede lograr el actuador neumático. Para la primera fase fue necesario diseñar una estructura portante encargada de almacenar el conjunto de piezas que serán la carga implantada en el actuador para tener un acercamiento del comportamiento. En la segunda fase fue indispensable implementar una herramienta que sujetará el sensor al banco de prueba, debido a que sin un soporte fijo este presentaba vibraciones que entorpecían el proceso de obtención de los datos.

La importancia de tener varias muestras es encontrar un promedio, buscando establecer una muestra de datos reales y filtrados, para así obtener una línea de tendencia adecuada y ajustada, para nuestro caso se recogieron 8 muestras por cada sentido de movimiento.

Una vez realizado el laboratorio (carga y descarga) del actuador neumático, se obtuvieron los datos desde el “puerto de visualización” del arduino; estos se copiaron en un archivo .csv y se convirtieron de texto a columnas como se muestra en la figura 7. Gracias a la plataforma Matlab® se importaron los datos mediante la función `csvread ('bajada.csv')`; después de tenerlos el workspace fueron graficados en un delta de tiempo con pasos de 0.6 segundos para verificar que los datos obtenidos fueran útiles como se muestran en las figuras 7.

Como se muestra en la figura 7 el dato # 5 tuvo que ser despreciado por que se aleja del comportamiento real y es tomado como una medición falsa e inadecuada. Una vez interpretados los datos, se filtraron bajo la forma de la media móvil para obtener una gráfica realista y promediada como se muestran en las figura 7.

Descripción de la prueba

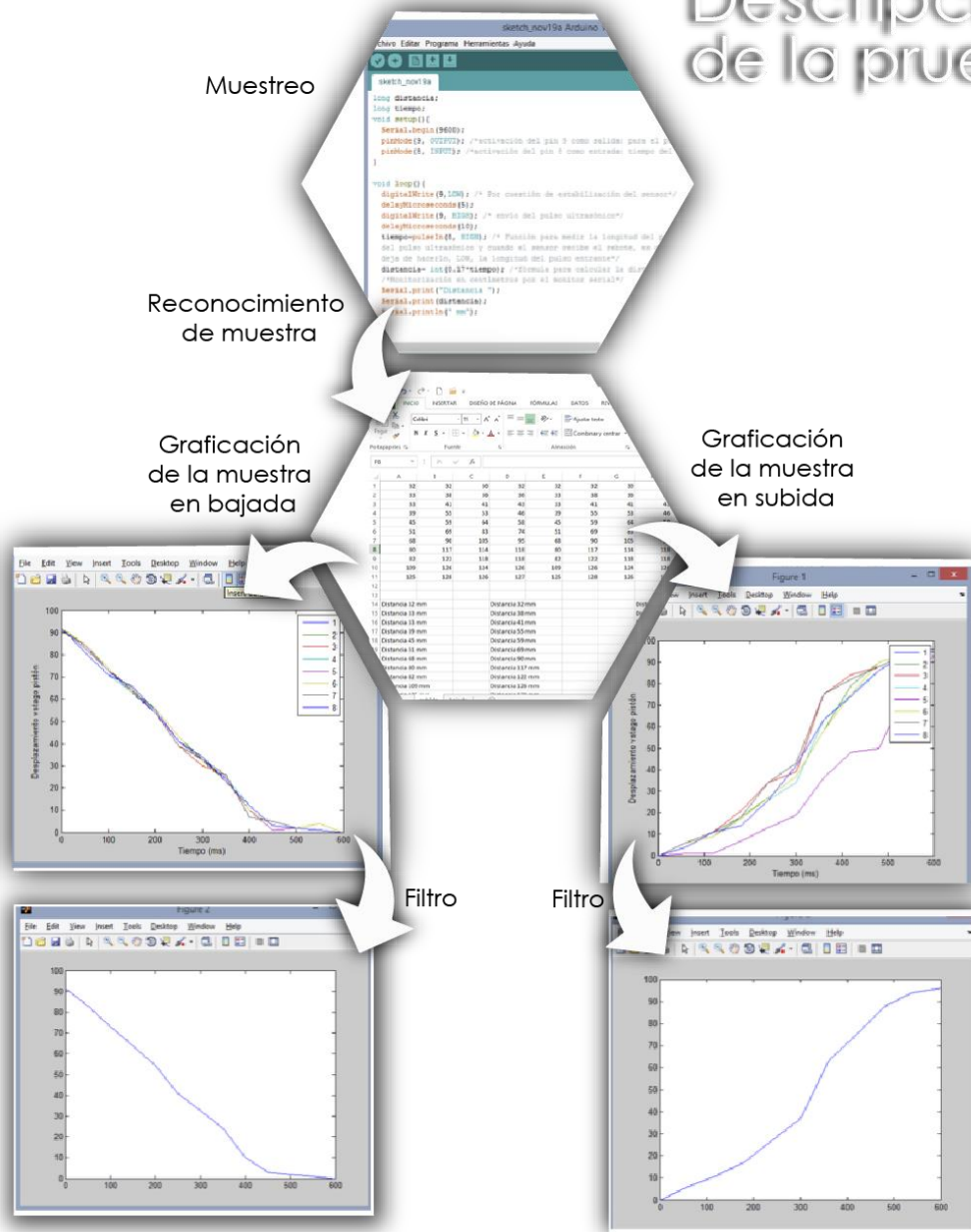


Figura 7, Descripción de la prueba.

Los anteriores datos fueron exportados al programa Excel para obtener mediante una de sus herramientas la línea de tendencia de cada una de las interacciones continuado de la ecuación característica de las mismas como se muestra en las figura 8. Posteriormente se importa a la plataforma Matlab® que mediante toolbox “ident” quien es el encargado de estimar la función de trasferencia que representa el funcionamiento característico del actuador neumático.

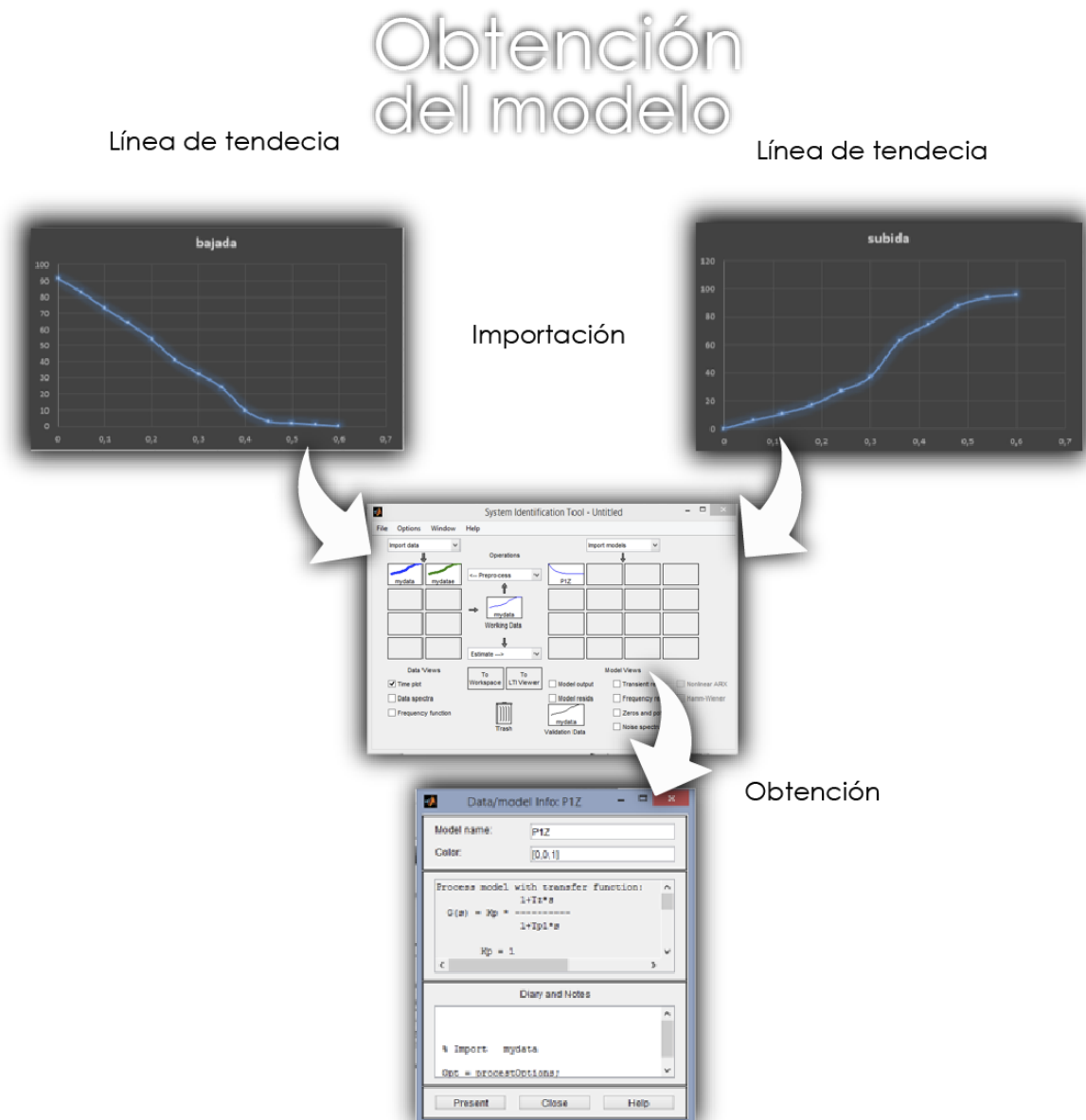


Figura 8, obtención del modelo.

El método de identificación adoptado para la obtención de la función de transferencia es el de respuesta en frecuencia utilizando dos polos y un cero, debido a que por el comportamiento del sistema tal método es eficiente al realizar la estimación de la función de transferencia.

Una vez encontrada la función de transferencia se realiza la discretización cambiándola del dominio de la place al dominio de Z, tal es discretizada en este caso con un tiempo de muestreo de 0.01 s. más tarde se realiza una evaluación del sistema a un escalón unitario. Como se muestra en la figura 9, cabe aclarar que el método de orden zero es escogido gracias a que se puede evidenciar que la función de transferencia tiene la característica de ser de orden uno y este método facilita trabajo arrojando el mismo resultado.

Transformada Z

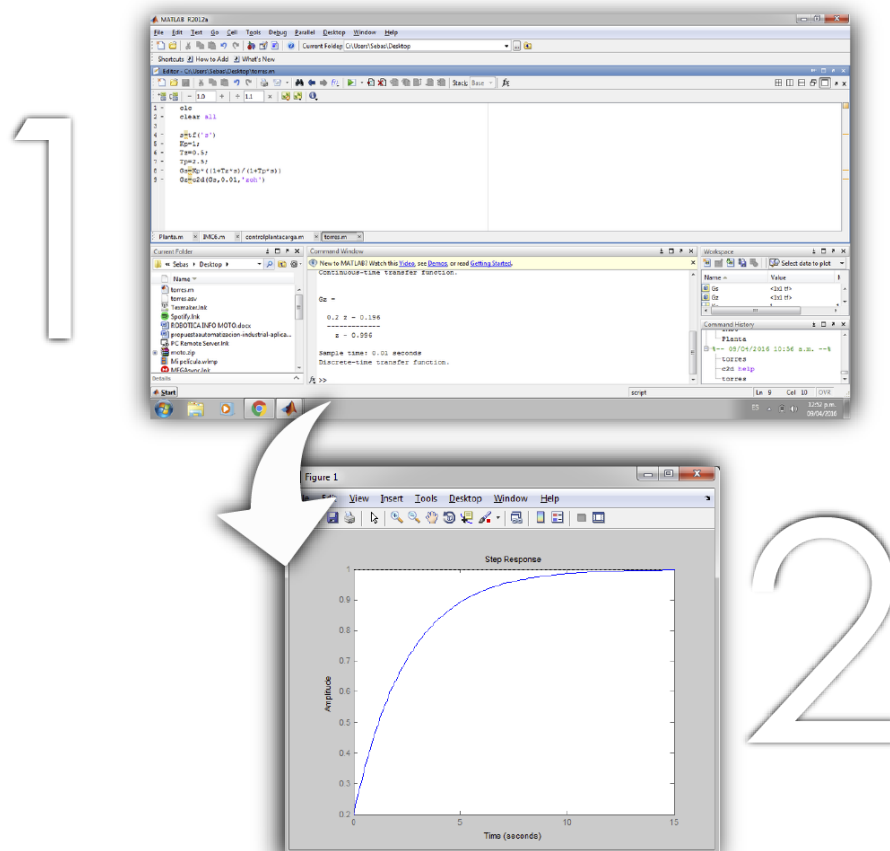


Figura 9, transformada Z.

Al contener la función discretizada se procede a operar algebraicamente la ecuación (2.17) bajo los parámetros de tiempo (2.18), para establecer la ecuación en diferencias que describe la salida de la planta que para este caso es el pistón como se muestra en la ecuación (2.21).

$$\frac{y(z)}{x(z)} = \frac{0,2 z - 0,196}{z - 0,996} \quad (2.17)$$

$$t = 0,01 \text{ s} \quad (2.18)$$

$$y(z) - z^{-1}y(z)0,996 = 0,2x(z) - 0,196z^{-1}x(z) \quad (2.19)$$

$$y(z) = 0,2x(z) - 0,196z^{-1}x(z) + z^{-1}y(z)0,996 \quad (2.20)$$

$$y(k) = 0,2x(k) - 0,196x(k - 1) + 0,996y(k - 1) \quad (2.21)$$

Es importante aclarar que la ecuaciones en diferencia reflejan el comportamiento del actuador neumático teniendo en cuenta la instrucción generada por el usuario, así mismo dicha está caracterizada por reconocer la muestra anterior (posición anterior) y la muestra actual (posición actual), generando la diferencia respecto al valor entregado por el usuario en su estrategia de control.

2.2.3. SELECCIÓN DE COMPONENTE

En este caso se busca establecer cualidades y desventajas de tres posibles actuadores, como se muestra en la tabla 5, es importante aclarar que se mencionó que los fabricantes de estas máquinas estipulan un método único por su facilidad de en instalación, costo y rapidez en la interacción.

Con el objetivo inicial de este proyecto, según experiencias, cotizaciones en compañías fabricantes y consejos de mecánicos, se puede establecer más claramente las ventajas de los actuadores neumáticos frente a los demás (mantenimiento, instalación y su costo inicial), dejando al descubierto sus características técnicas para suplir los requerimientos mecánicos y físicos de este proyecto; uno de los parámetros más significativos es su valor comercial frente a los demás, ya que al pensar ubicar un inversionista en el futuro se optaría por tener optimizados los costos de fabricación.

Tabla 5. Comparación de actuadores

| Características | Tipo de sistema | | |
|-----------------|--|--|---|
| | Neumático | Hidráulico | Eléctrico |
| Ventajas | Precisión Temperatura Limpieza Velocidad Regulación sobrecargas | Fuerza Vida útil Flexibilidad Elemento reversible Regulación | Precisión Regulación Fácil control Silenciosos |
| Desventajas | Preparación Obtención Ruidos Velocidad Costo fuente | Velocidad Limpieza Alta presión Costo | Potencia Costo |
| Costo aprox | 30.000 a 560.000 | 120.000 a 950.00 | 73.000 a 1.500.000 |

Adicionalmente se realizó un seguimiento sobre los precios ofrecidos en el mercado colombiano y para obtener un punto referencial económico como se muestran en los anexos.

2.2.4. ELECTROVÁLVULA

El dispositivo como su nombre lo indica es una válvula electrónica como se muestra en la figura 10, su modo de operación es de (0 a 24v o 0 a 115v) para el fabricante festo, dicha es la encargada de recibir un valor digital y su salida virtual se reflejara en función (0 o 6 bar), valores que serán objeto de uso en la plataforma virtual, otorgándole al pistón un correcto funcionamiento en sus dos posibles estados. En este caso se busca una electroválvula de nomenclatura 3/2 de auto retorno

mecánico donde al suministrarle voltaje permite el flujo de presión y al cortar el suministro de voltaje retorna a su estado inicial denegando el flujo de presión.



Figura 10, Electroválvula 3/2. [25]

2.3. PROPUESTA DE AUTOMATIZACIÓN MESAS ROTACIONALES

Por último se empleará un motor eléctrico en la parte inferior de máquina, otorgando un acople por medio de tornillos y sus respectivas tuercas; dicho motor será el encargado de dar movimiento rotacional a las mesas que se muestran en la figura 11, buscando dar giros de 60° (entre brazo y brazo), donde si recordamos aquellas mesas originalmente gracias a su rodamiento ubicado en la parte central permite un movimiento libre suministrado por el operario. La posterior conexión será eléctrica llegando a un variador que será el encargado de modificar su frecuencia para otorgar al motor una protección a la hora de arranque y su ulterior parada, aquel también tendrá como objetivo permitir el posicionamiento adecuado de las mesas.

²⁵ [25]Festo 2016.https://www.festo.com/net/SupportPortal/Files/17269/Ventile_es.pdf.

Es de gran importancia recalcar que al igual que el pistón esta propuesta es sugerida por los fabricantes debido a su bajo costo y fácil implementación, adicionalmente esta propuesta compuesta por un motor eléctrico, puede ser calculada de forma orgánica para establecer la capacidad o clase a utilizar.

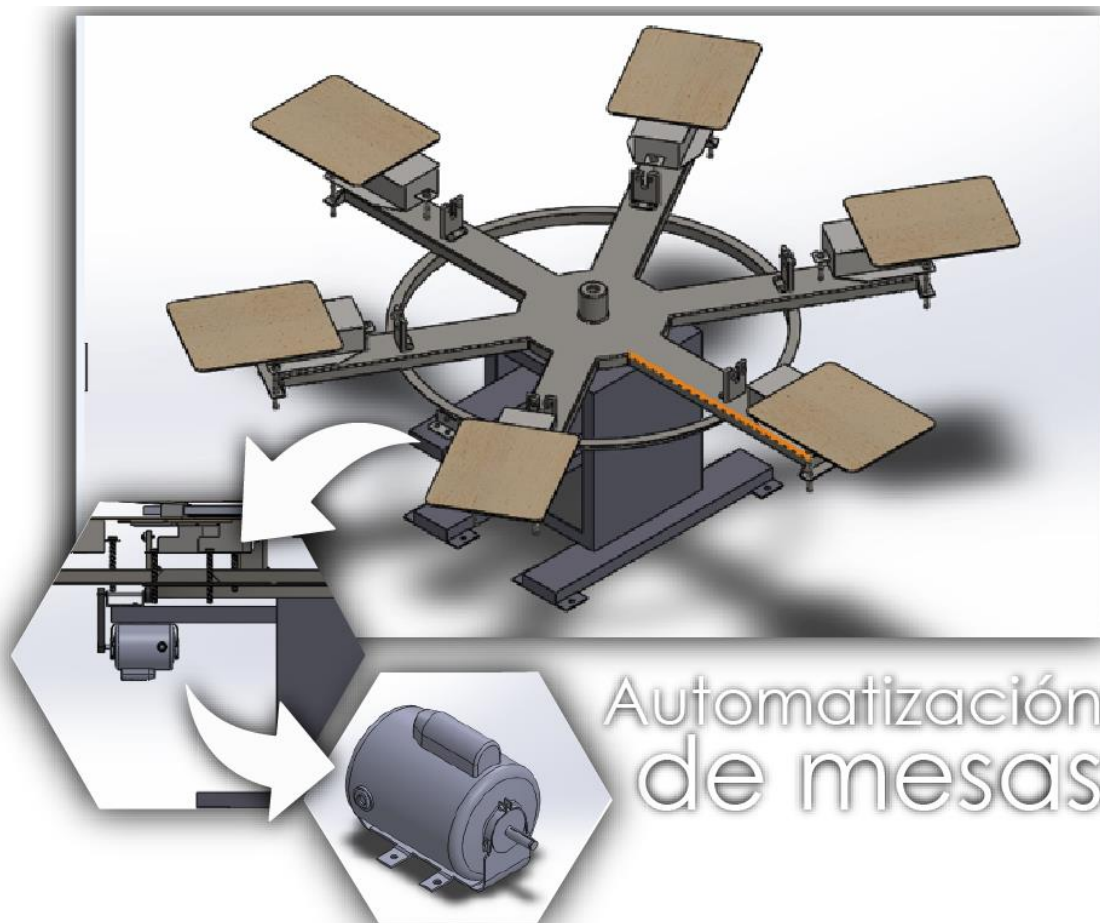


Figura 11, Automatización de mesas.

2.3.1. CÁLCULOS MATEMÁTICOS PARA DETERMINAR EL MOTOR

Para la generación de los cálculos fue necesario utilizar el programa Solidworks donde se implementaron las propiedades de masa y momento de inercia, conociendo que dicho programa tiene la capacidad de establecer estos parámetros como se muestra en la Tabla 6; es adecuado aclarar que los elementos involucrados en la automatización para producir el movimiento rotacional son únicamente las mesas mostradas en la figura 12, debido a que su estado pre-automatizado contenía un movimiento libre y con un centro de rotación ubicado en el eje principal donde se concentran las fuerzas de las mesas y brazos.

Tabla 6, Propiedades de las mesas

| Propiedades de masas de las mesas | | | |
|---|--------------------|--------------------|-------------------|
| | X | Y | Z |
| Centro de masa | 0 mm | 673.55 mm | 0 mm |
| Momentos de inercia (g*mm) | Ix=(0,0,1) | Iy=(1,0,0) | Iz=(0,1,0) |
| Momentos de inercia (g*mm) | Px=(9484919021.93) | Py=(9484919021.23) | Izz=(18942847486) |
| Masa: 47075.54 g | | | |
| Volumen: 47075540.49 mm ³ | | | |
| Área de superficie 3230881.62 mm ² | | | |

Conociendo la ecuación (2.22) que me relaciona la potencia del motor en función al peso de la estructura (mt), la resistencia de translación (w), el rendimiento mecánico (n) y la velocidad de translación (v).

$$P = mt * w * \frac{v}{2\pi * 9,55n} \quad (2.22)$$

Se realiza el cálculo del peso total de las mesas y se deja expresado en función de N el cual es multiplicado por la resistencia de translación, luego se multiplica por la velocidad de translación y se divide en $2\pi + 9,55$ por el rendimiento mecánico como se muestra en la ecuación (2.23).

$$47,1kg * \frac{9,81m}{s^2} = 462,01 N \quad (2.23)$$

La determinación de la velocidad de translación se ve reflejada sobre la ecuación (2.25) la cual es despejada de la ecuación (2.24). Se conoce que el ángulo de rotación es $\theta = 60^\circ$, el tiempo $t = 2$ segundos para ser aplicados en la ecuación (2.25).

$$w^2 = w_0 + 2\alpha\theta \quad (2.24)$$

$$w = \sqrt{2\alpha\theta} \quad (2.25)$$

Para establecer α = aceleración angular, se hace la equivalencia de la ecuación (2.26) y (2.27) entregando como resultado la ecuación (2.28) y al ser remplazadas por las variables nos entrega la ecuación (2.29)

$$\theta = \frac{\pi}{3} rad \quad (2.26)$$

$$\theta = \frac{1}{2} \alpha t^2 \quad (2.27)$$

$$\alpha = \frac{\theta}{\frac{1}{2} t^2} \quad (2.28)$$

$$\alpha = \frac{\pi}{6} rad \quad (2.29)$$

Una vez encontrada la aceleración angular se procede a remplazar las variables dentro de la ecuación (2.25) como se muestra en la ecuación (2.30) entregando como resultado la ecuación (2.31).

$$w = \sqrt{2 \left(\frac{\pi}{6} \right) \left(\frac{\pi}{3} \right)} \quad (2.30)$$

$$w = \frac{\pi}{3} rad \quad (2.31)$$

Después se procede a establecer la velocidad lineal caracterizada por la ecuación (2.32) y al convertir las variables como se muestra en las ecuaciones (2.33 y 2.34) nos arroja la ecuación (2.35) como resultado.

$$v = w * r \quad (2.32)$$

$$v = \frac{\pi rad}{3 s} * 1m \quad (2.33)$$

$$v = \frac{\pi m}{3 s} \quad (2.34)$$

$$v = 62,83 \frac{m}{min^{-1}} \quad (2.35)$$

Por último se remplaza las variables en la ecuación (2.22) para arrojarnos como resultado la ecuación (2.36), al realizar un cambio de variables nos arroja como resultado la ecuación (2.37).

$$P = 0,4232KW \quad (2.36)$$

$$P = 0,5675hp \quad (2.37)$$

Al encontrar como resultado la potencia el KW, su posterior conversión nos establece un resultado de 0.56 hp que al buscar comercialmente un valor en potencia de motores eléctricos nos implanta el valor como se muestra en la ecuación (2.38).

$$P \approx 1hp \quad (2.38)$$

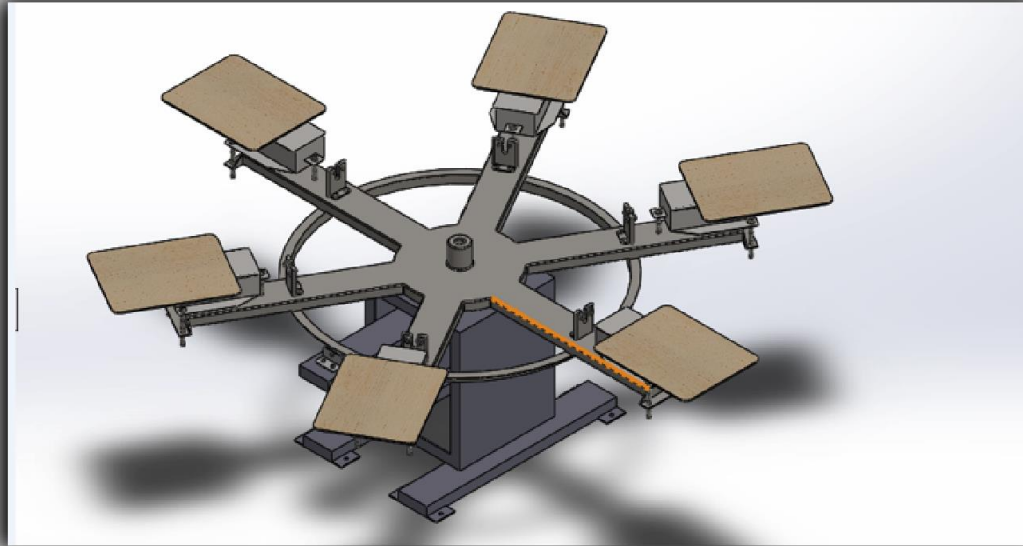


Figura 12, Mesas

2.3.2. MODELAMIENTO MATEMÁTICO DEL MOTOR ELÉCTRICO

Como se conoce un motor de corriente continua como un actuador electromecánico, que por medio de una excitación eléctrica suministrada en los polos produce repeler con una frecuencia los imanes sujetos al motor, produciendo un movimiento rotacional. Es de gran importancia aclarar que dicho motor contempla un modelo más sencillo que facilita el trabajo y que puede entregar las mismas prestaciones que un motor de corriente alterna, adicionalmente se buscó establecer ciertas constantes bajo parámetros del fabricante baldor donde se evidencia la resistencia del bobinado, masa del rotor y su respectivo diámetro.

El procedimiento de modelado mostrado en esta sección sigue el desarrollo visto en [25].

Como se muestra en la ecuación (2.39) la cual refleja velocidad del rotor en función del tiempo, adicionalmente está representada por una tensión aplicada al inducido, dos constantes derivadas del torque γ β , y por ultimo un delta de tiempo. Más tarde el autor refleja como se muestra en la ecuación (2.40), la función de transferencia que caracteriza el motor en función de la velocidad angular y el voltaje.

$$w = \frac{v_i}{\beta} x \left(1 - e^{-\frac{\beta}{\gamma} t} \right) \quad (2.39)$$

$$G(s) = \frac{w(s)}{v(s)} = \frac{k}{s * t + 1} \quad (2.40)$$

Una vez se tiene la función de transferencia (2.40) se procede a identificar las variables (k y t), las cuales se describen como se muestran en las ecuaciones (2.41 y 2.42)

$$k = \frac{1}{kb} \quad (2.41)$$

$$t = \frac{R * J}{Kp * Kb} \quad (2.42)$$

Donde la constante Kb está en función del voltaje según la ley Lenz como se muestra en las ecuaciones (2.43), al reorganizar la ecuación anteriormente mencionada nos arroja como resultado la ecuación (2.44).

$$Vi = Kb * w \quad (2.43)$$

$$Kb = \frac{Vi}{w} \quad (2.44)$$

El momento de inercia J , se expresa en la ecuación (2.45), de la cual depende directamente la constante Kp como se ve en la ecuación (2.46)

$$J = \frac{1}{2} m r^2 \quad (2.45)$$

²⁶ [25] www2.uca.es/grup-invest/ntgc/crealabcp/practicas/scap1.pdf

$$Kp = \frac{j * \alpha}{I} \quad (2.46)$$

Una vez encontradas las variables se procede a establecer el t y k reemplazando en los valores anteriormente involucrados dejando como respuesta las ecuaciones (2.47) y (2.48).

$$k = 2.66 \quad (2.47)$$

$$t = 3.3 \quad (2.48)$$

Al contener la función discretizada se procede a operar algebraicamente la ecuación (2.49) bajo los parámetros de tiempo (2.50), para establecer la ecuación en diferencias que describe la salida de la planta que para este caso es el motor como se muestra en la ecuación (2.52).

$$\frac{y(z)}{x(z)} = \frac{0,008048}{z - 0,997} \quad (2.49)$$

$$t = 0,01 \text{ s} \quad (2.50)$$

$$y(z) - z^{-1}y(z)0.997 = 0,008048x(z) \quad (2.51)$$

$$y(z) = 0,0080408x(z) + z^{-1}y(z)0,997 \quad (2.52)$$

$$y(k) = 0,0080408x(k) + 0,997y(k - 1) \quad (2.52)$$

El procedimiento de la obtención de la función de transferencia se ve reflejada en los anexo 11 donde se puede evidenciar paso a paso el proceso para la obtención de los modelos.

2.4.2 ACTUADOR PARA MOVIMIENTO ROTACIONAL DE LAS MESAS

Para satisfacer la necesidad mecánica y tener un control eficaz se buscó implementar un motor cc de 110v, conociendo que existe una importancia en la precisión que se necesita en el proceso, que por costos y facilidad de control frente a un posible motor neumático es el designado para este proyecto. Para ello se realizó una comparación netamente comercial de los productos ofrecidos en el mercado colombiano, como se muestra en la tabla 7.

Tabla 7. Comparación de motores comerciales

| Compañía fabricante | Motor asincrónico de 1 HP | | |
|---------------------|---------------------------|---------------------------|-------------|
| | Valor comercial | Ventajas | Desventajas |
| Siemens | 445.200 | Garantía Mantenimiento | Costo |
| ABB | 379.000 | Mantenimiento | Garantía |
| Baldor | 510.000 | Garantía | Costo |

2.4.3. VARIADOR DE FRECUENCIA

El dispositivo como su nombre lo indica es un variador de frecuencia como se muestra en la figura 13, en función de voltaje donde se ingresa un valor digital y su salida es netamente análoga por sus variaciones frecuenciales, otorgándole al motor un correcto funcionamiento de encendido, parada y para nuestro caso giros de 60°. En este caso se buscó un variador que actúe con voltajes 110 vcc, adicionalmente el dispositivo debe ser capaz de controlar la posición del motor sincrónico dándole la facilidad al usuario de generar pasos de 60° (movimiento de mesa en mesa). Las señales de control que este dispositivo recibe en el entorno virtual serán de tipo digital (0 o 24v), que son las variables que le darán la apertura al "circuito" y su posterior corte para la generación de movimientos controlados. Adicional es importante resaltar que este variador tiene la capacidad de frenado casi instantáneo produciendo un error mínimo (-3 a +3 °), pero al pensarlo no será ningún inconveniente puesto a que su acople mecánico permitirá la corrección de la misma.

Pensando en el control de posicionamiento del motor se buscó un variador que tuviera fácil implementación y del cual se tuviera un conocimiento mínimo previo, para esto propuso el variador varitel de referencia V65D debido a sus características técnicas (voltajes y corrientes), permitiendo un control eficaz sobre el motor y trayendo como resultado el perfecto funcionamiento virtual y real de la máquina de estampado.



Figura 13, variador de frecuencia. [26]

2.5. DETERMINACIÓN DE LA RED NEUMÁTICA

2.5.1. COMPRESOR PARA LA RED NEUMÁTICA

La respuesta a cotizaciones de solo compresores de funcionamiento tipo “tornillo” más conocidos técnicamente como rotativos, es debido a su eficiencia y costo frente a las demás formas de funcionamiento como se muestra en la tabla 8, optimizando costos de mantenimiento de la red neumática y del mismo, obviamente el costo inicial es mayor pero si realizamos un centro de costos encontraremos como resultado la reducción drástica dando objeción adecuada a la inversión inicial.

²³ [26] Variador de frecuencia. Varitel. Variador DC. <http://www.varitel.com/assets/images/DC-drives.jpg>

Tabla 8. Tipos de compresores

| Compresores | | |
|----------------|---------------------------------|-------------------------|
| Tipo | Ventajas | Desventajas |
| Embolo | Costo | Eficiencia garantía |
| Rotativo | Eficiencia, costo y garantía | Mantenimiento |
| Turbocompresor | Garantía | Costo, mantenimiento |

2.5.2. CALCULO CAPACIDAD DE UN ACUMULADOR

Para nuestro caso con un único actuador neumático, se busca satisfacer una presión constante de 6 bares y una capacidad mínima que se ve reflejada en la ecuación (2.45), donde el volumen del depósito está en función de una constante k y el caudal del compresor. El caudal es convertido en las variables adecuadas para operar como se muestra en la ecuación (2.46) y al remplazar en la ecuación inicial nos arroja como resultado la ecuación (2.47), entregándonos una capacidad mínima como se muestra en la ecuación (2.48).

$$v = k * Q \quad (2.45)$$

$$Q = 6 \text{ bar} = 0.7151 \frac{m^3}{min} \quad (2.46)$$

$$v = 0.4 * 0.7151 \quad (2.47)$$

$$v = 0.2860 m^3 \quad (2.48)$$

Una vez encontrada la capacidad mínima se establece una capacidad de 150 litros según requerimientos por el cliente desfasándonos mucho más de lo requerido con la intención de una futura ampliación, esto es estipulado según conocimientos técnicos y la ficha técnica del actuador donde se ve reflejado que su consumo máximo por operación es de 0.3 L como se muestra en los anexos , adicionalmente se planteó un esquema de la línea de distribución con sus respectivos accesorios (líneas de servicio, separador o filtro), como se muestra en la figura 14

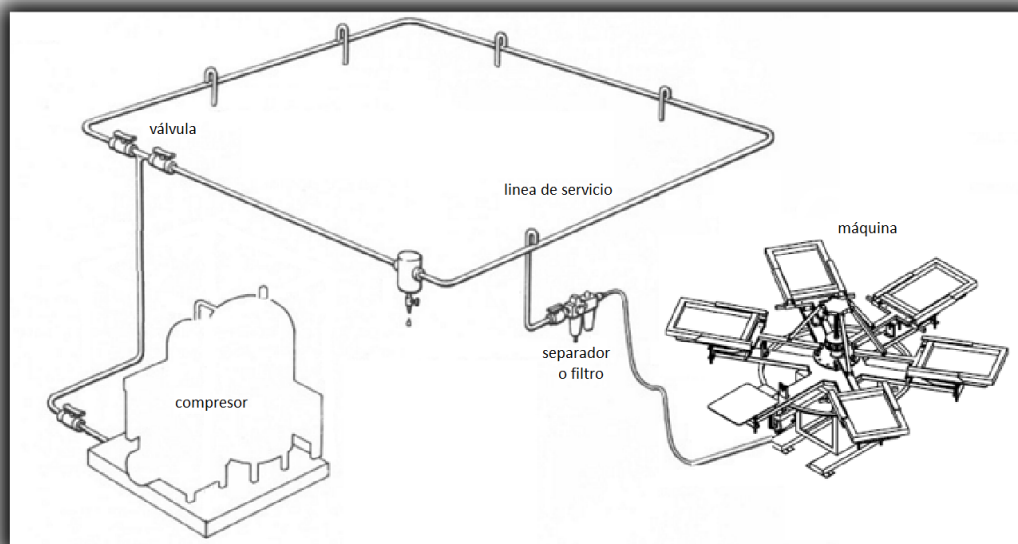


Figura 14, Modelo de red neumática

3. VIRTUALIZACIÓN DE LA MÁQUINA EN UNITY

Este capítulo describe el diseño, modelamiento, programación y virtualización de la interfaz gráfica de la máquina de estampado.

3.1. MODELADO DE LA MÁQUINA

La creación de la máquina en un entorno visual fue dado gracias al programa Solidworks® optando por todas sus características y facilidades de uso en un entorno 3D como se muestra en la figura 15. Es de gran importancia establecer que el diseño de esta máquina parte desde la base física y la propuesta de automatización generada, se describe como un proceso de diseño de piezas de forma independiente y gracias a herramientas contenidas en el programa se replicaron el número de piezas necesarias; su ensamblaje fue dado de la base hacia las mesas dejando como punto central el eje de rotación contemplado en el eje Y. Como se aprecia en los anexos 6, 7 y 8 se puede entrar en detalle sobre las características de diseño implementadas en este proyecto.

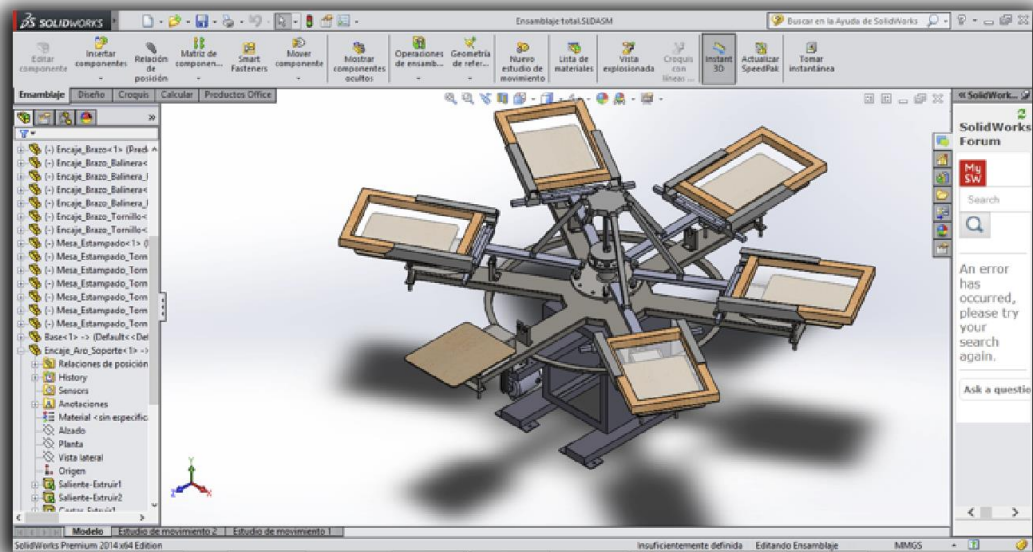
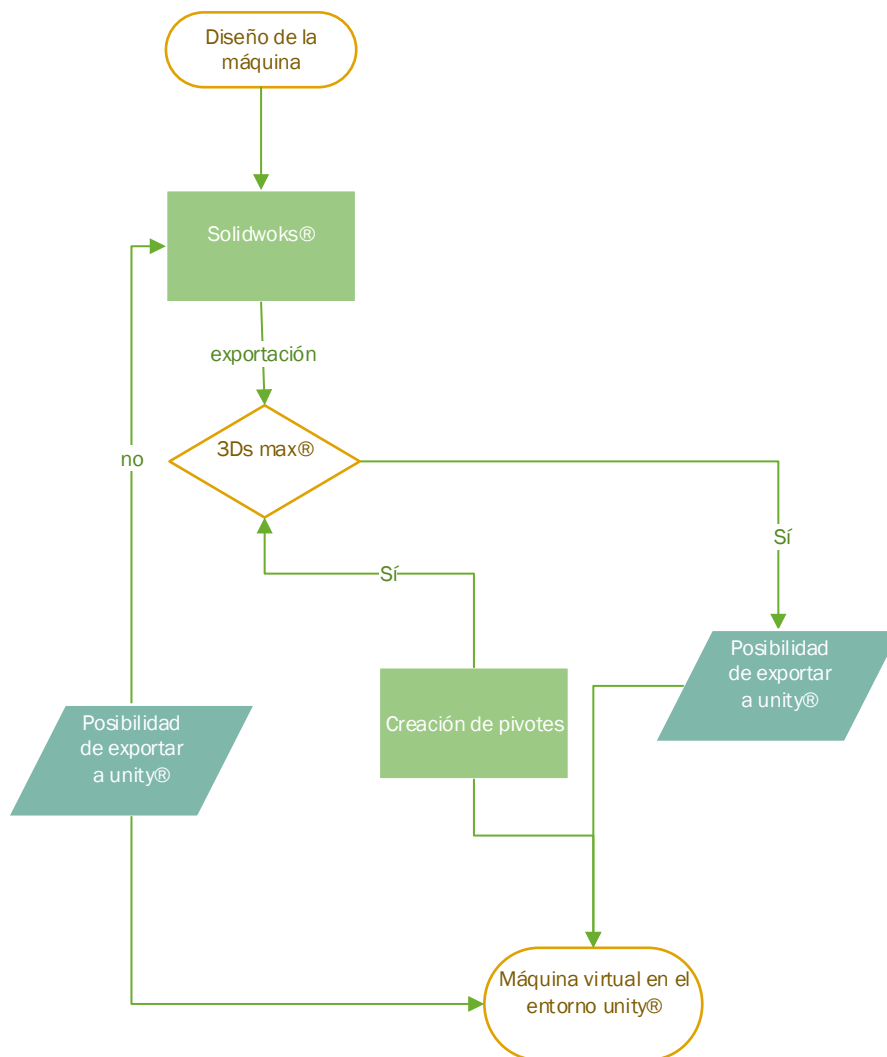


Figura 15, Modelado de la máquina en Solidworks.

Adicionalmente se puede evidenciar en el diagrama de flujo la estrategia de diseño que fue utilizada para la creación de la máquina virtual, donde fue necesario utilizar dos plataformas para la generación adecuada y eficaz de la interfaz gráfica.



Posteriormente se exportó al programa 3Ds max® como se muestra en la figura 16, el cual otorga todas las características de un programa 3D reconociendo pieza a pieza y dejando como resultado el ensamble de la máquina, que permitió su ulterior exportación al programa Unity®, ya que desde la plataforma Solidworks® no se permite exportar al entorno Unity®.

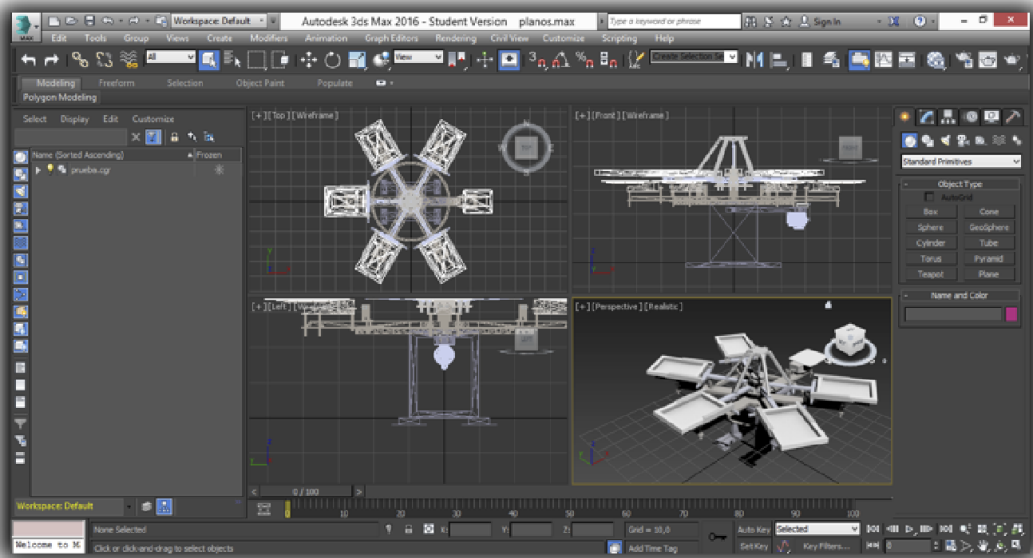


Figura 16, Modelado de máquina en 3Ds max®.

Fue necesario generar pivotes ya que la plataforma unity® no tiene la capacidad de reconocer los puntos de acopio o pasadores que fueron diseñados de la máquina en la plataforma Solidworks®, la creación de los pivotes para generar el movimiento vertical encadenado de los brazos es dado a través de la plataforma 3Ds max®, ya que el movimiento es otorgado por el pistón sobre el acople de los brazos en sentido vertical y cada uno de los acoples y extensiones debe corresponder encadenadamente ante la interacción otorgada por el usuario. Luego se exportó al programa unity que por medio de una de sus extensiones (final IK), se realiza el proceso de reconocimiento de cada uno de los pivotes fijados en 3Ds max® dado en los pasadores que unen las extensiones permitiendo así el movimiento adecuado de la sección de los brazos, como se muestra en a figura 17.

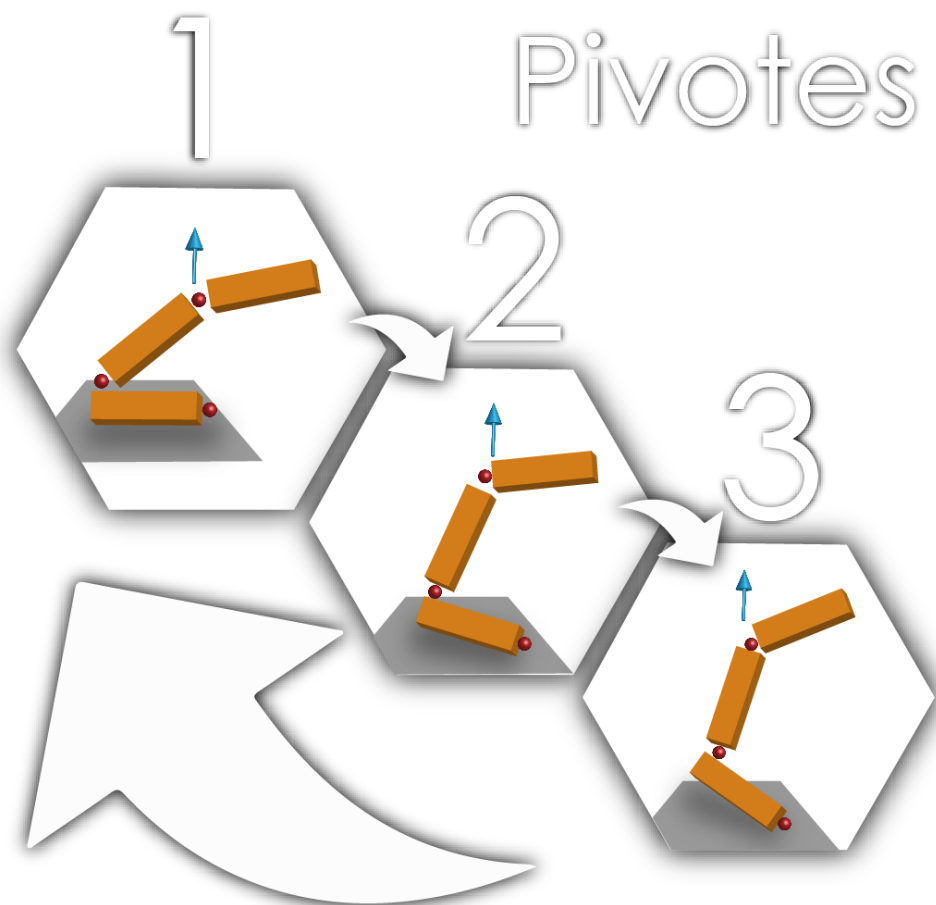


Figura 17, Principio de los pivotes.

3.2. IMPORTACIÓN A UNITY®.

Para llevar a cabo la importación desde la plataforma se agregó el archivo .FBX usando la herramienta obtenida al dar un clic derecho, dentro del espacio de los assets como se muestra en la figura 18, posteriormente se arrastró con un clic a la escena que se estaba creando, obteniendo así la visualización de la máquina como se muestra en la figura 19. Tal importación tiene la capacidad de reconocer el despiece de cada una de las partes que conforman la máquina de estampación, dejando que el usuario manipule cada una de ellas y obtenga posteriormente una animación adecuada sobre cada una de las mismas.

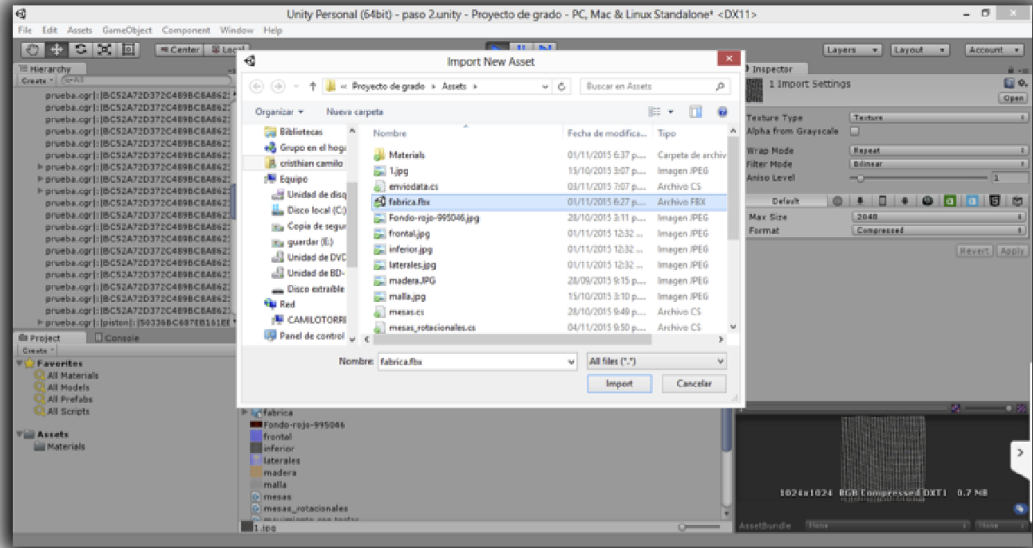


Figura 18, Importación de un asset.

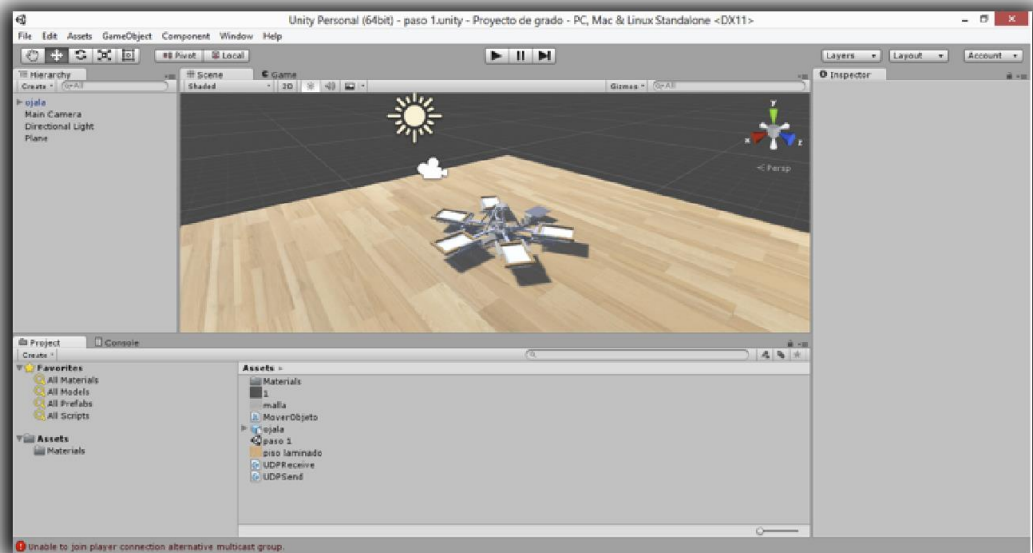


Figura 19, Asset visualizado en la escena principal.

3.4.1 MOVIMIENTO ROTACIONAL

Después de su correcta importación se inició la concatenación de las piezas dependiendo de su movimiento (rotacional y lineal sobre el eje Y), para ello se arrastró fragmento a fragmento a una pieza principal, donde luego se implementó un script bajo el lenguaje C# que provocara el movimiento en cada caso. La primera prueba que se realizó fue la rotación de las mesas en una escena del entorno Unity® como se muestra en las figura 20.



Figura 20, Movimiento rotacional.

3.3 INTERFACES DE ENTRADA A UNITY®.

La importancia sobre el conocimiento e inclusión de las adecuadas y reales variables de entrada y salida es la futura implementación a un hardware o incluso a la propia máquina automatizada, para nuestro caso (motor y pistón), tales pasaran por las siguientes 4 etapas:

- Control
- Dispositivo que controla el mecanismo
- Modelo matemático
- Simulación

Se presenta adicionalmente el código bajo el lenguaje c# que genera las instrucciones de movimiento según los datos resultantes de la comunicación, la inclusión de los modelos matemáticos del motor y del pistón buscando generar una réplica del comportamiento real de cada dispositivo anteriormente nombrado como se muestra en los anexos 9 y 10.

3.3.1. PISTÓN

El control del pistón se basa en la interacción neta del usuario a partir de los modelos matemáticos para incursionar en la creación de una estrategia de control bajo parámetros (0 a 24 v), que son los valores reales que la electro válvula recibe para generar la conmutación de salida o entrada de presión (0 a 6 bar), según corresponda. Posterior a la electro válvula dichos valores entraran en el modelo matemático anteriormente encontrados para la recreación más aproximada del movimiento, por último dichos valores de salida serán (1 a 3), donde el modelo matemático será el encargado de generar según la programación del escenario virtual la simulación detallada del proceso requerido por el usuario. Como se muestra en la figura 21.

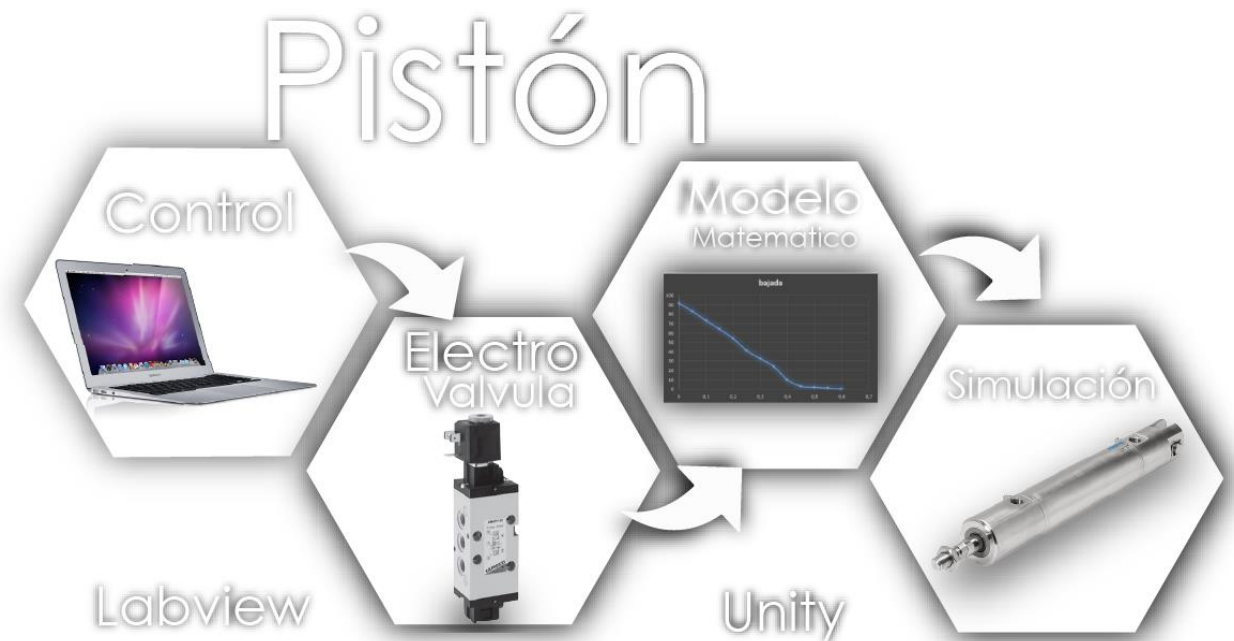


Figura 21, Interfaz de comunicación del pistón.

La creación de un control de lazo cerrado es una de las estrategias más usadas actualmente por que busca adaptar la planta bajo el error encontrado y para eso es necesario que la plataforma Unity® mediante la programación del código genere los valores de estados (arriba o abajo), en el cual podría estar el pistón, luego debe enviarlos hacia la plataforma Labview® para su retroalimentación, dejando parámetros para una adecuada estrategia de control.

3.3.2. MOTOR

El control del motor se basa en la interacción neta del usuario a partir de los modelos matemáticos para incursionar en la creación de una estrategia de control bajo parámetros (0 o 24v), que son los valores reales que el variador de frecuencia recibe para generar la conmutación de arranque o parada en función de (0 a 110vcc), según corresponda. Posterior al variador dichos valores entraran en el modelo matemático anteriormente encontrado para la recreación más aproximada del movimiento, por último dichos valores de salida (0 a 11500), del modelo matemático serán los encargados de generar según la programación del escenario virtual la simulación detallada del proceso requerido por el usuario. Como se muestra en la figura 22.

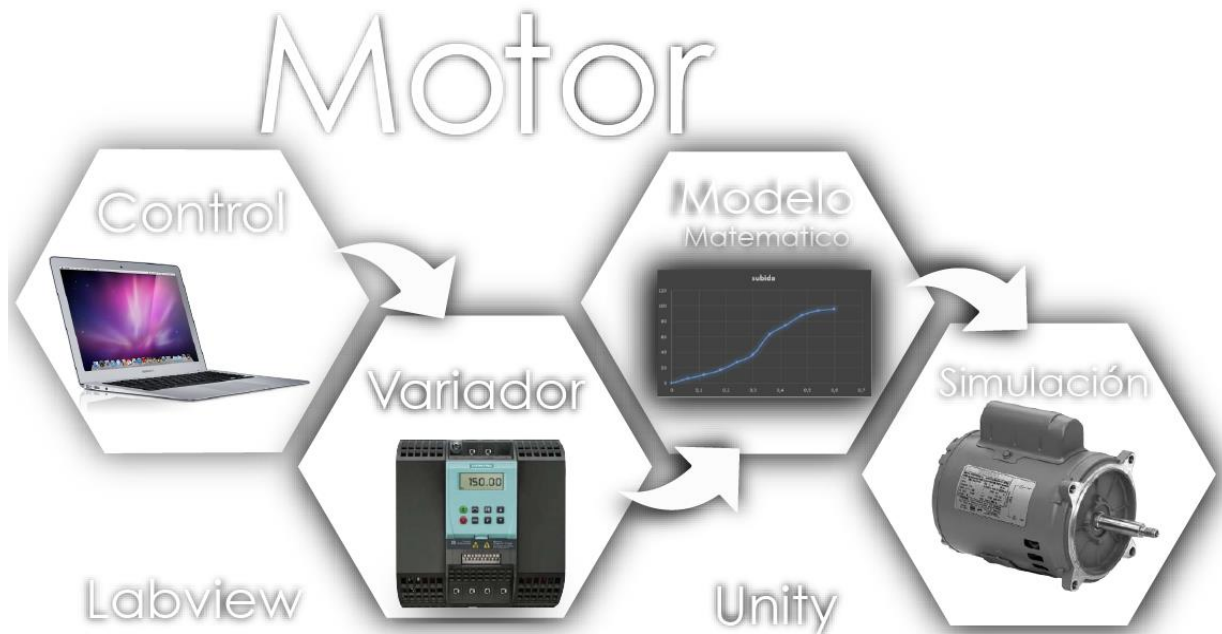


Figura 22, Interfaz de comunicación del motor.

Como anteriormente se nombró un control de lazo cerrado es una de las estrategias más versátiles y para este caso también fue necesario el envío del valor de la posición en la cual se encuentran las mesas en valores reflejados (0 a 360), luego debe enviarlos hacia la plataforma Labview® para su retroalimentación, dejando parámetros para una adecuada estrategia de control.

3.5 PROGRAMACIÓN DE MOVIMIENTOS

Conociendo anteriormente que labview® es capaz únicamente de enviar datos concatenados en forma de strings, Unity® los recogerá como valor string y los convertirá en forma de enteros, posteriormente serán los valores aplicados a las ecuaciones diferenciales dependiendo de cada caso, adicional unity® tendrá la capacidad de separar el arreglo de datos recibidos y convertirlos en variables globales.

3.5.1 MOVIMIENTO VERTICAL

La forma adecuada para la producción del movimiento comienza desde el diseño de la máquina, debido que si no se le establecen los pivotes desde el inicio a la estructura, cuando la máquina virtual se encuentre en la plataforma Unity® no podrá obedecer a una interacción concatenada. El principio básico de un pivote es el encadenamiento de movimientos, donde al aplicar un movimiento a X pieza seguidamente provoque un movimiento sobre Y pieza es decir como se muestra en la figura 16. Al realizar le procedimiento correcto en el programa 3Ds max® y su adecuada importación como anteriormente se ha nombrado, se sitúa un script bajo el lenguaje C# que provocara el movimiento encadenado de los brazos en ambos sentidos como se muestra en las figura 23.



Figura 23, Movimiento vertical.

4. COMUNICACIÓN LABVIEW® CON UNITY®

Este capítulo describe la implementación de la comunicación, desarrollo y programación obtenida para el control adecuado de la máquina virtual. Es importante recalcar que se está llevando una simulación sobre un solo equipo hardware y la comunicación generada es netamente interna (virtual), debido que este proyecto no busca interactuar entre dos o más equipos una simulación y su debido control, su objetivo es generar una estrategia de software in the loop.

4.1. COMUNICACIÓN UDP LABVIEW® A UNITY®

Para producir la comunicación entre el entorno visual y labview® (programa que permite la interacción programable con el usuario), fue necesario implementar un tipo de comunicación UDP asociado con el protocolo ip donde se le suministro un puerto (8051) y una dirección ip (127.0.0.1). Una de las pruebas piloto fue el envío de la palabra (Universidad Piloto de Colombia) como se observa en la figura 24.

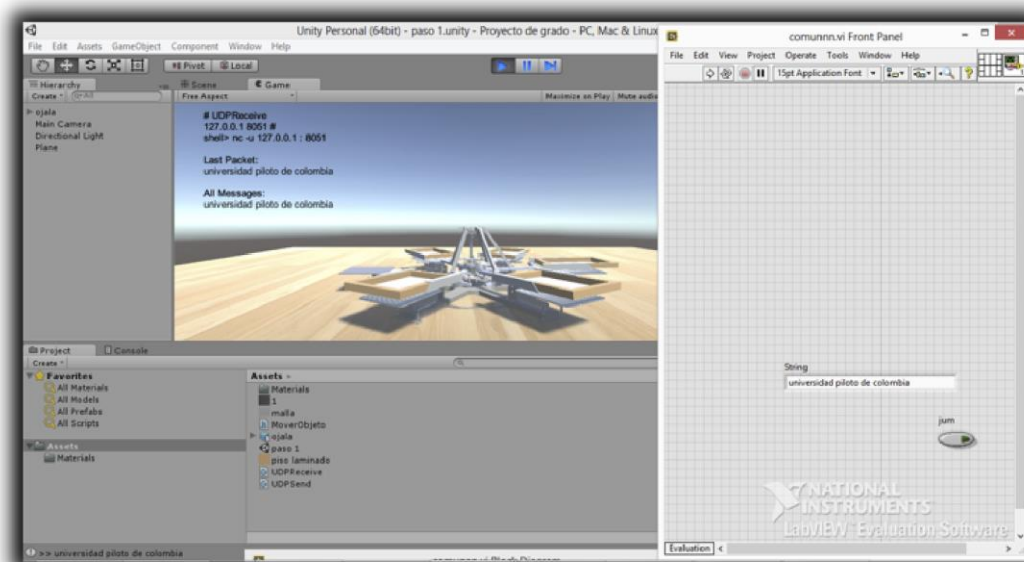


Figura 24, Prueba piloto de comunicación.

El desarrollo de la comunicación en la plataforma labview® inicia desde la estipulación de un ciclo while, posterior se ingresaron de la paleta de funciones los

componentes que generan la apertura del puerto, el envío o recepción y el cierre de la comunicación, una vez generados los conectores de los tres componentes se procede a insertar un componente string el cual es el encargado de almacenar las variables de entrada a la plataforma o la variable de salida de la misma.

La generación de la comunicación dentro de le entorno unity®, es dado mediante un ciclo de instrucción fijando una ip y un puerto el cual debe ser programado de igual forma en la plataforma labview®, debido a que la lógica de este tipo de comunicación es la generación de un servidor virtual a las cuales las dos plataformas debes ser conectadas de forma inmediata. Es de gran importancia recalcar que sobre este tipo de comunicación solo se puede enviar datos en forma de string según parámetros de la plataforma Labview®. Conociendo que dentro de la comunicación bajo el parámetro ip existen perdidas de datos, se buscó empaquetar las variables de forma que se asegure que ninguna de ellas sea despreciada o perdida en cada ciclo de comunicación, aclaremos que las variables que se desean enviar son separadas por carácter en banco (espacio), para que más tarde dentro de la plataforma de visualización grafica se puedan decalar como variables independientes.

Luego de esto, por medio de la comunicación establecida entre los dos programas, se envió un valor que se reflejaría en la velocidad del movimiento rotacional de las mesas de estampado como se muestra en la figura 25.

1 Comunicación

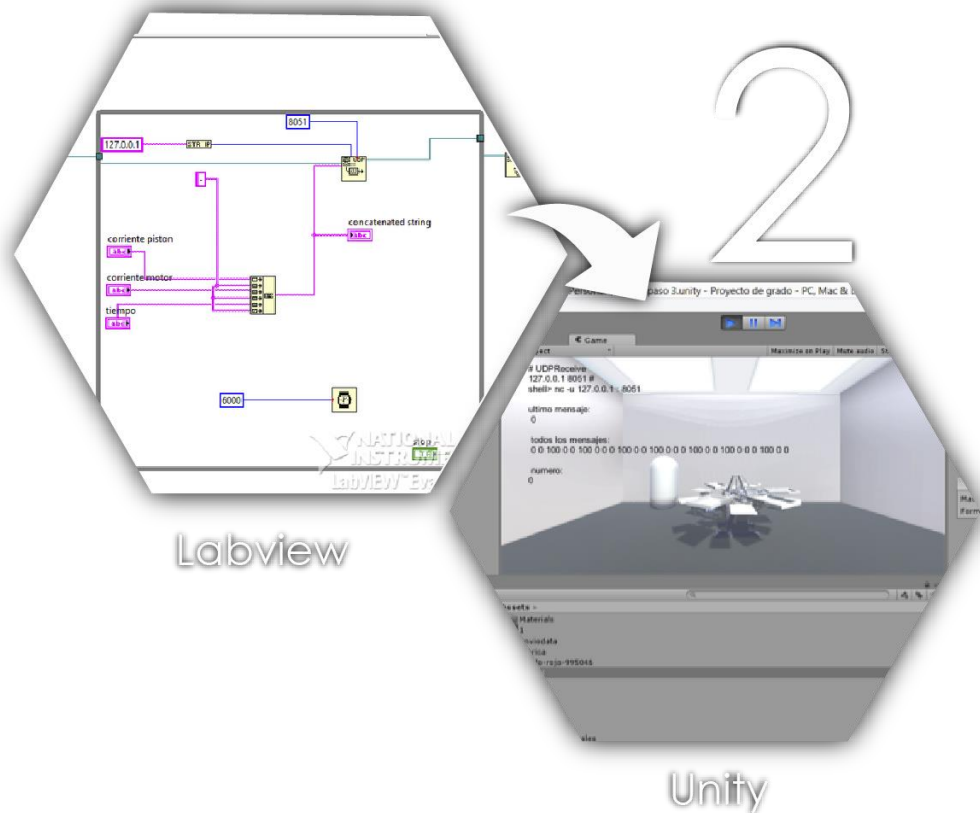


Figura 25, Comunicación.

Después de corroborar que existe entrada de datos a la plataforma unity, se empaquetan los datos desde labview® para que unity® los reciba, los desencadene y los opere.

Adicionalmente se puede estipular que la programación bajo el lenguaje c# como se muestra en los anexos 9 y 10 contienen una instrucción de tipo while donde se espera el parámetro string el cual es el enviado desde la plataforma de control, luego se convierte en un array para ser recorrido dentro de una instrucción, dejando como variables independientes enteras el arreglo anteriormente mencionado. Una vez generadas las variables enteras se procede a entrar a un ciclo de tipo while el cual actualiza el estado actual del objeto y el estado anterior del mismo, para ser

multiplicado por ciertas constantes dejando como salida un valor de ajuste para cada una de los objetos virtuales.

Dentro de la rutina principal se establecen dos parámetros físicos de la máquina, ya que al detener su correcta manipulación el usuario debe tener las posiciones físicas de los componentes a controlar en lazo cerrado (pistón y motor). De esta forma se entrega al usuario instantáneamente, después de recibir la cadena de caracteres de inicio, los valores físicos en la cual se encuentra.

4.2. COMUNICACIÓN UDP UNITY® A LABVIEW®

Al igual que la anterior se establece un puerto (8060) y una dirección ip (127.0.0.2). Una de las pruebas piloto fue el envío de la palabra (cfour sas). Luego de confirmar que existe el envío de datos desde la plataforma unity®, se empaquetan datos en un string desde unity® para que luego lab view los reciba, los desencadene, los convierta en valores enteros y los opere. En la parte de la recepción fue necesario dividir los valores y guardarlos en variables. El diagrama de flujo de ésta rutina y para el Programa Principal del Software es la que se muestra en la figura 26.

Comunicación

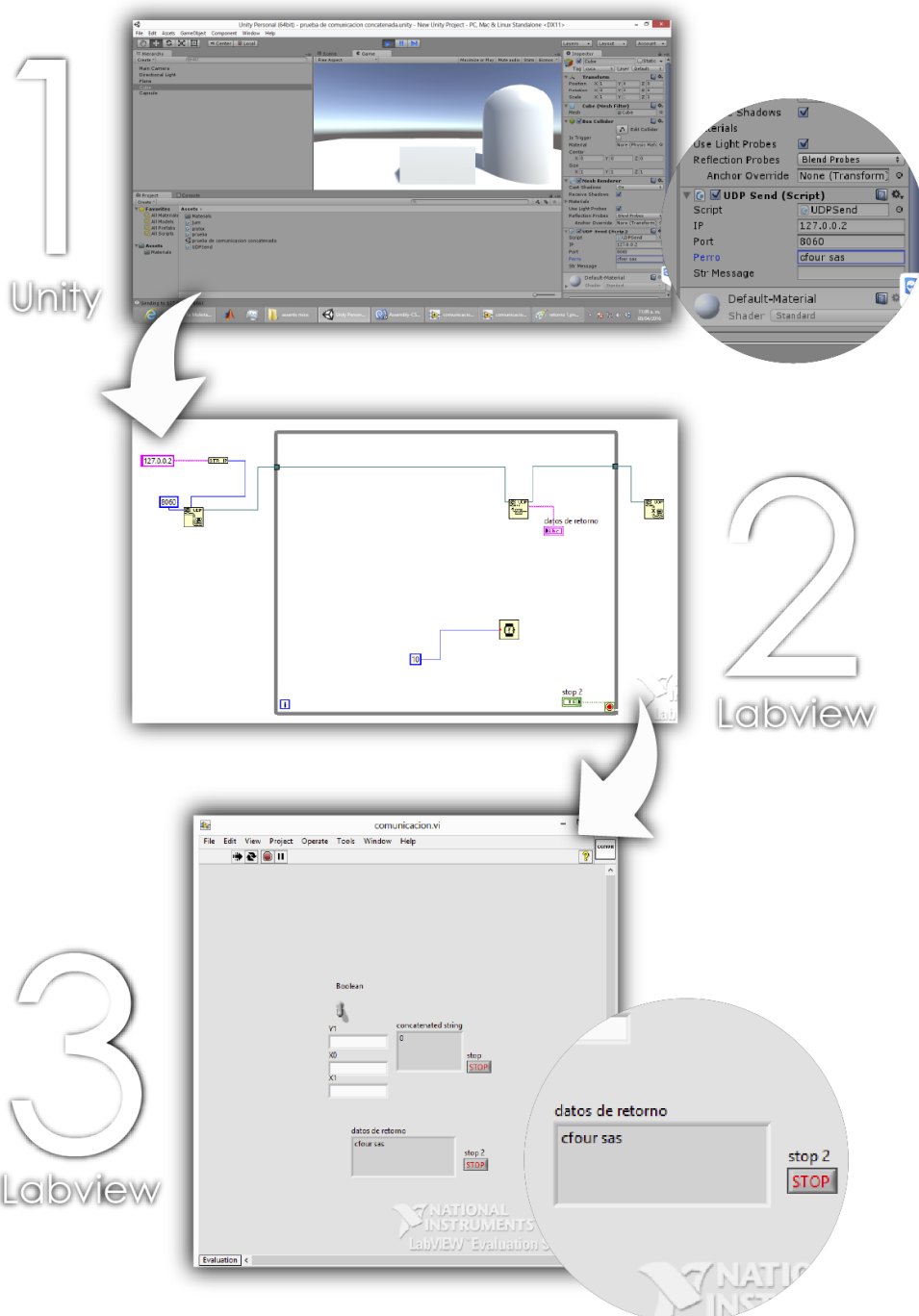


Figura 26, Comunicación retorno

4.3. APLICACIÓN DEMOSTRATIVA DE LA MÁQUINA

Se tiene claro el concepto de control de lazo cerrado, el cual es un sistema de retroalimentación que tiene variables (para este caso sensores), estos alimentan inmediatamente, después de realizar el proceso, la señal de entrada provocando la optimización y correcto funcionamiento del proceso como se muestra en la figura 27.

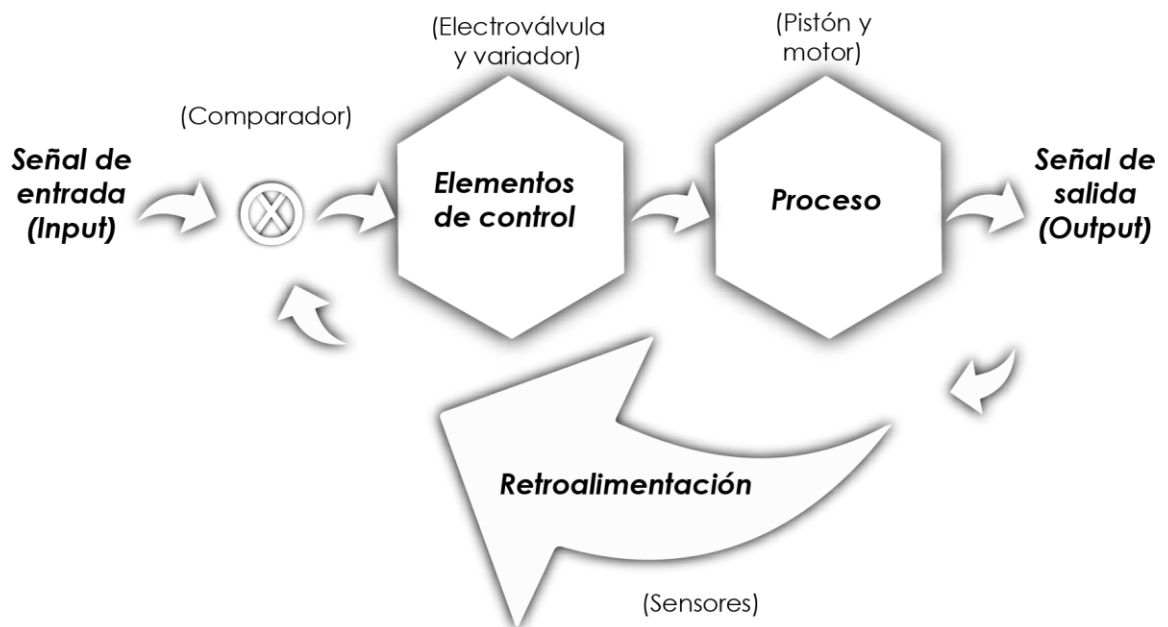


Figura 27, Modelo de Comunicación.

La creación del sistema de control fue dado gracias a la implementación de una simulación del sistema en el programa matlab® quien gracias a uno de sus toolbox “simulink” como se muestran en las figuras 28 y 29, permitió la estimación de un pid bajo los parámetros de la planta anteriormente buscados (ecuación en diferencias), es importante aclarar que la estimación fue dada gracias a la función “auto - sintonización” que nos permite establecer de forma eficaz e inmediata los parámetros proporcionales, integrativos y derivativos.

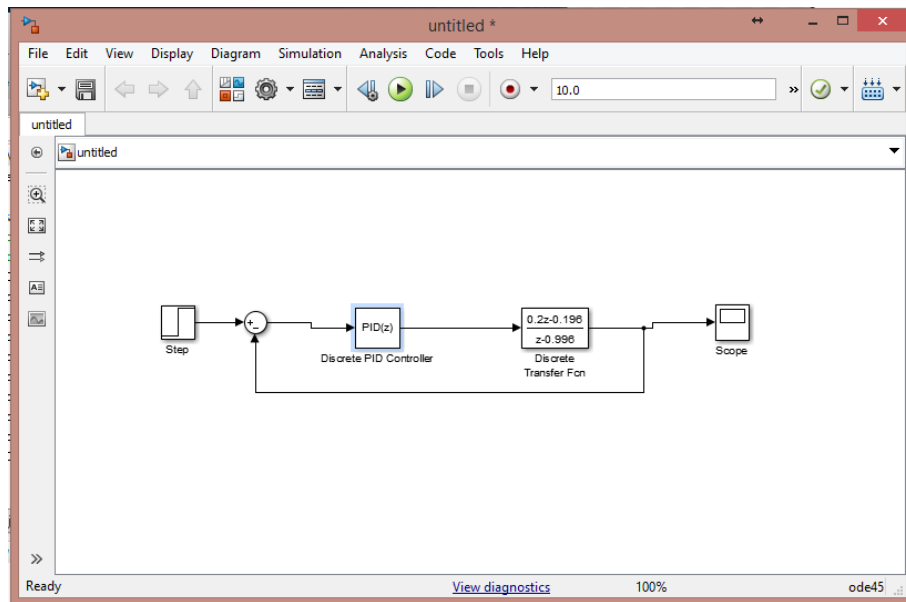


Figura 28, Simulink pistón.

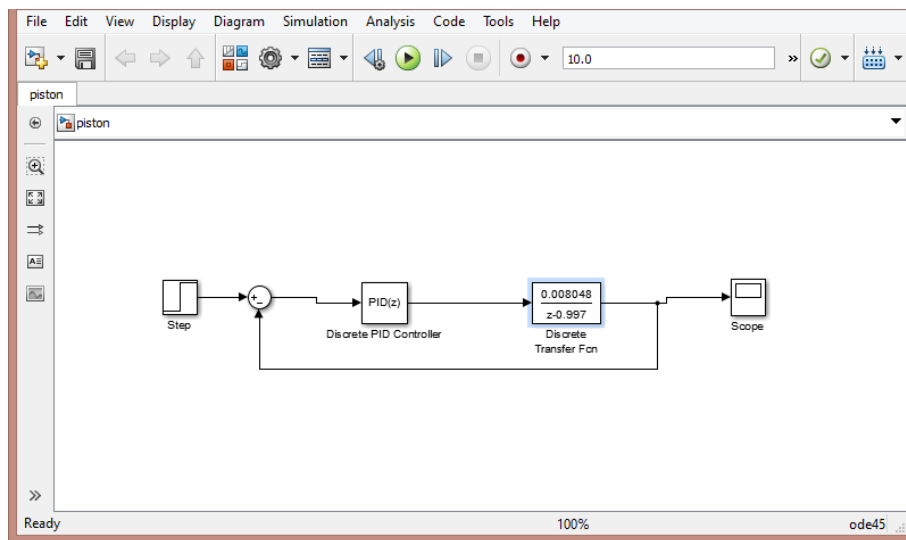


Figura 29, Simulink motor.

Una vez procesada la ecuación en diferencia para el pistón y motor, este arroja como resultado una gráfica que denota el comportamiento del pid frente a la planta, donde el usuario puede evidenciar el delta de tiempo que necesita para estabilizar el sistema. Como se muestra en la figuras 30 y 31.

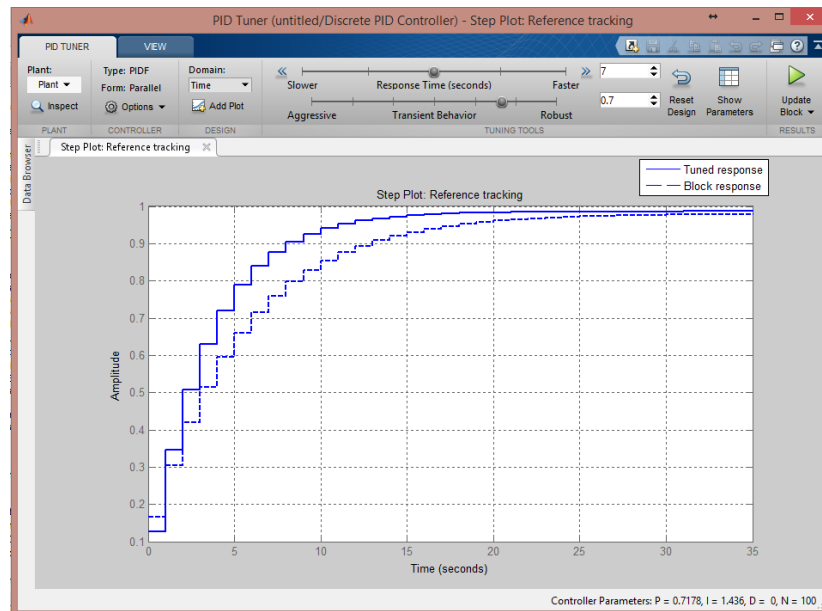


Figura 30, comportamiento del pid pistón.

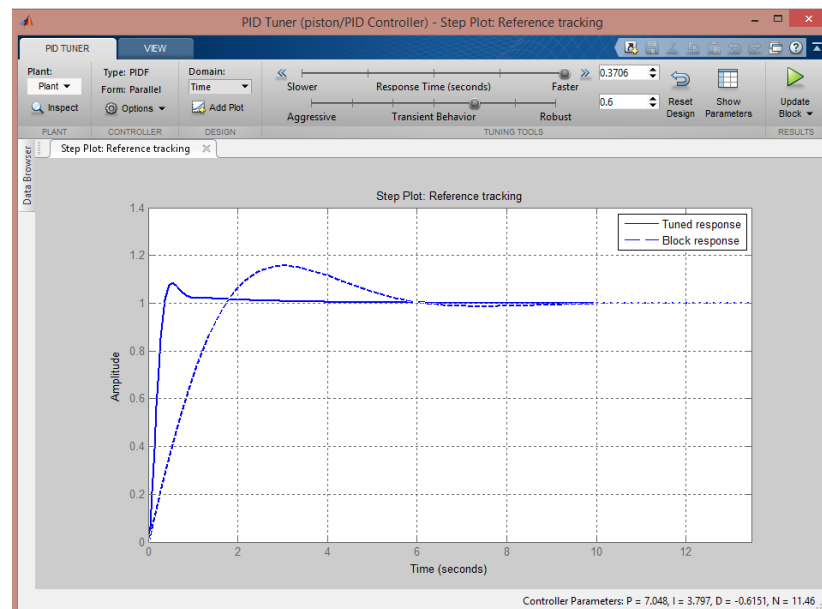


Figura 31, comportamiento del pid motor.

Se encuentra parámetros de control en el modelo del motor un $P = 7.048$, $I = 3.797$ y $D = -0.6151$, adicionalmente se establece que los parámetros del pistón son $P = 0.7178$, $I = 1.435$, $D = 0$ siendo los parámetros que se ingresaran posteriormente en la interfaz de control. La aplicación demostrativa inicia desde el proceso de comunicación entre la plataforma Labview® y el motor de juegos Unity®. Como se nombra en el capítulo 4.2. Ésta interacción se da por medio del protocolo UDP, donde se crea un puerto y una IP independiente para el envío y otro para la recepción de los string. Así que, los datos enviados según parámetros de control

por el usuario desde la plataforma de Labview® en forma de string los recibe el motor de juegos (Unity®), el cual los decodifica generando una Array que al pasar por una instrucción de programación recorre y genera variables enteras.

Una vez que son interpretadas las variables enteras, entran a los modelos matemáticos que les entrega un valor para su posterior simulación. El envío de datos de la plataforma de Unity® hacia Labview® en primera instancia se genera por medio de código, se realiza la identificación del target del objeto que para el caso del pistón se busca establecer la posición ubicada en el eje Y, y en el de motor se busca la rotación orientada hacia el eje Y entregando unos valores vectoriales de los cuales fueron transformados en formato string para su posterior envío. Labview® lo reconoce como variables string y le practica una conversión para ser usados como variables independientes enteras. Como se muestra en la figura 32

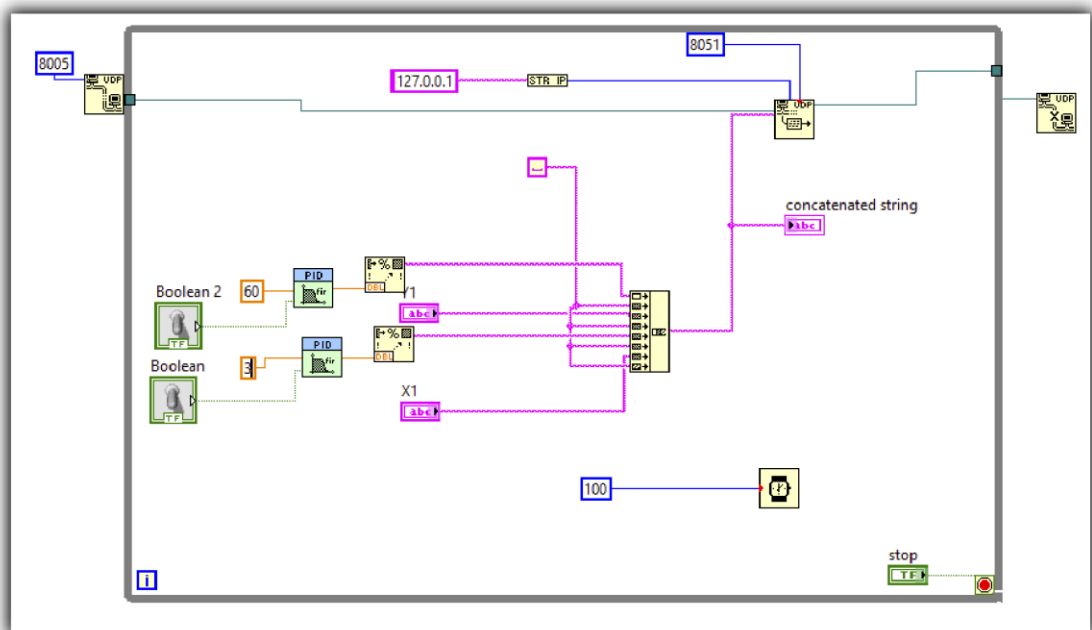


Figura 32, Fase labview

La demostración final de la plataforma donde se puede evidenciar su correcto funcionamiento del proceso de comunicación y control es como se muestra en la figura 33.

Aplicación demostrativa

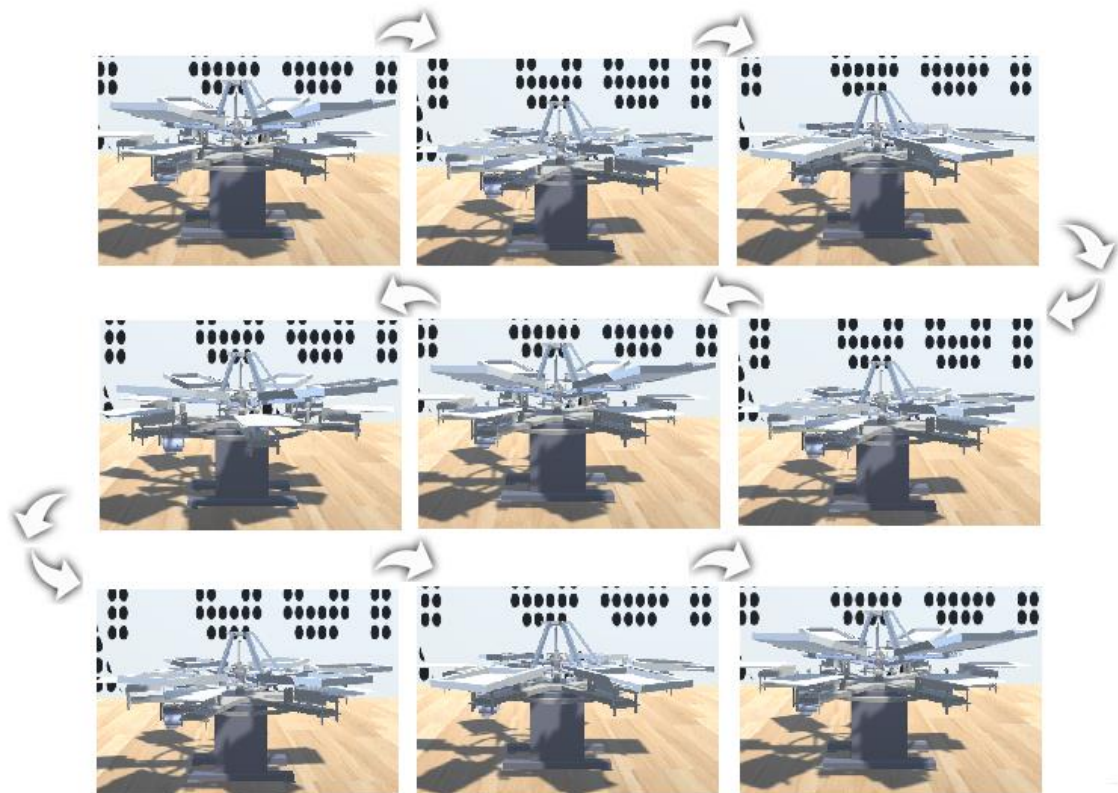


Figura 33. Aplicación demostrativa

5. CONCLUSIONES

En este capítulo consta de todos los resultados obtenidos del diseño, implementación y simulación de la máquina virtual de estampación. Además del funcionamiento de la comunicación (estrategia de software in the loop).

- La determinación de los elementos que permiten la automatización de la máquina fueron objetivo de respuesta de la necesidad del cliente, posteriormente se tuvo en cuenta las propiedades físicas (masas y momentos de inercia) de la máquina, para su respectivo análisis.
- El modelo matemático que define el comportamiento del pistón, se determinó por medio del modelo de un filtro cuyos datos (posición / delta de tiempo) fueron adquiridos experimentalmente, para llegar a si a un modelo en ecuaciones en diferencia y su implementación adecuada en el motor de juegos.
- Una de las formas eficientes para la creación de una aplicación que permita la interacción es por medio de la plataforma Lab view por ser una plataforma grafica que se caracteriza por su versatilidad en procesos educativos.
- Se realizó la comunicación con el protocolo IP/TCP (UDP) debido a que la perdida de datos no es muy relevante, se empaquetaron los datos en forma de strings debido a que las plataformas labview y unity se comunican correctamente de forma bidireccional.
- La validación del prototipo fue dado en un 90% pues no fue posible generar la actualización en tiempo real de la posición en la cual se encontraban los actuadores del modelo mecánico virtual de la máquina, debido a que el tiempo de respuesta es muy largo.

6. RECOMENDACIONES

- Se requiere optimizar los tiempos en los que la máquina virtual realiza retorno de valores al software de control para poder realizar un control más eficiente.
- Buscar optimizar la interfaz gráfica en la fluidez para que la aplicación de los modelos matemáticos sea correcta y no retarde el comportamiento normal de la máquina virtual.
- Es importante que los puertos de uso para esta aplicación (8050 y 8060) estén libres por todas las aplicaciones en el momento de inicio para no entrar en conflicto con los mismos.
- Es necesario que el estudiante entre en contacto con una máquina de estampado mecánica antes de construir una estrategia de control sobre la máquina de estampado textil automatizada.
- El usuario debe recordar que el PID puede ser modificado a su gusto o puede usar el PID determinado que se diseñó en el programa.

BIBLIOGRAFÍA

[1] RILEY, Patrick F.; RILEY, George F. Next generation modeling III-agents: Spades---a distributed agent simulation environment with software-in-the-loop execution. En Proceedings of the 35th conference on Winter simulation: driving innovation. Winter Simulation Conference, 2003. p. 817-825.

[2] KESAVADAS, T., 1995. Virtual Reality Based Interactive Automation of a Robotic Die Grinding Operation. Order No. 9531955 ed. Ann Arbor: The Pennsylvania State University.

[3] LOR, L.A., 2013. Supervisory Controller Validation for a Plug-in Parallel-through-the-Road Hybrid Electric Vehicle by Software-in-the-Loop Testing. Order No. 1543709 ed. Ann Arbor: Wayne State University ISBN 9781303312649.

[4] MAHIEU, F., 2000. Software Development of a Hardware-in-the-Loop Simulation and a Three-Dimensional Viewer for Autonomous Underwater Vehicles. Order No. 1398140 ed. Ann Arbor: Florida Atlantic University ISBN 9780599650664; 0599650664.

[5] KANG, Y., et al, 2013. Design of Driver-in-the-Loop Simulator for Off-Highway Truck. Sensors & Transducers, 12, vol. 161, no. 12, pp. 401-407 ProQuest Computing; ProQuest Science Journals; ProQuest Telecommunications. ISSN 23068515.

[6] CARR, D.A., 2009. Towards Automatic Parallel Game Engine Architectures. Order No. 1464428 ed. Ann Arbor: University of Nevada, Reno ProQuest Dissertations & Theses A&I. ISBN 9781109145281.

[7] CARRILLO, M.D., 2013. VizLab: The Design and Implementation of an Immersive Virtual Environment System using Game Engine Technology and Open Source Software. Order No. 1539751 ed. Ann Arbor: The University of Texas - Pan American ProQuest Dissertations & Theses A&I. ISBN 9781303157882.

[8] ZHOU, T., 2008. Accurate Depth Based Post-Processing for Perception Enhancement in Real Time Three-Dimensional Graphics. Order No. 3305913 ed. Ann Arbor: George Mason University ProQuest Dissertations & Theses A&I. ISBN 9780549527701.

[9] 2014. GM Global Technology Operations LLC; Patent Issued for Vehicle Simulation System with Software-in-the-Loop Bypass Control. Computers, Networks & Communications, Jan 02, pp. 2938 ProQuest Career and Technical Education; ProQuest Computing; ProQuest Science Journals.

- [10]COLLINS, D.E. and GEORGE, A.D., 2001. Parallel and Sequential Job Scheduling in Heterogeneous Clusters: A Simulation Study using Software in the Loop. *Simulation*, Nov, vol. 77, no. 5, pp. 169 ProQuest Science Journals. ISSN 00375497.
- [11]KOSTÁL, P. and MUDRIKOVÁ, A., 2011. USE OF E-LEARNING AND VIRTUAL LABORATORY TO AUTOMATION TEACHING. *Annals of the Faculty of Engineering Hunedoara*, vol. 9, no. 4, pp. 183-186 ProQuest Research Library. ISSN 15842665.
- [12]KESAVADAS, T., 1995. Virtual Reality Based Interactive Automation of a Robotic Die Grinding Operation. Order No. 9531955 ed. Ann Arbor: The Pennsylvania State University ProQuest Dissertations & Theses A&I, <http://www.ni.com/labview/esa/>.
- [13]HUGHES, S.B., 2005. Assisted Viewpoint Interaction for 3D Visualizations. Order No. 3192962 ed. Ann Arbor: University of Pittsburgh ProQuest Dissertations & Theses A&I. ISBN 9780542355523; 0542355523.
- [14]HAN, S.H., 2010. Automated Post-Simulation Visualization of Modular Building Production Assembly Line. Order No. MR67935 ed. Ann Arbor: University of Alberta (Canada) ProQuest Dissertations & Theses A&I. ISBN 9780494679357.
- [15]BARKA, N., DESCHÊNES, J. and BROUSSEAU, J., 2013. Laboratory Activities in Industrial Sequential Automation through a Virtual and Physical Environment. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 04, vol. 3, no. 2, pp. 85 ABI/INFORM Global; ProQuest Education Journals. ISSN 20103654. DOI <http://dx.doi.org/10.7763/IJEEEE.2013.V3.198>.
- [16]NEUGEBAUER, J.-., 1992. Virtual Reality - More than just Simulation. *The Industrial Robot*, vol. 19, no. 3, pp. 30 ABI/INFORM Global; ProQuest Research Library; ProQuest Science Journals. ISSN 0143991X.
- [17]MCLELLAN, Hilary. Virtual realities. *Handbook of research for educational communications and technology*, 1996, p. 457-487.
- [18]CASIS, E., et al. Virtual automation. *Clinical leadership & management review: the journal of CLMA*, 2000, vol. 15, no 2, p. 89-91.
- [19]REINHART, Christoph; BRETON, Pierre-Felix. Experimental validation of Autodesk® 3ds Max® Design 2009 and DAYSIM 3.0. *Leukos*, 2009, vol. 6, no 1, p. 7-35.

[20]ELLIOTT, Chance, et al. National instruments LabVIEW: a programming environment for laboratory automation and measurement. Journal of the Association for Laboratory Automation, 2007, vol. 12, no 1, p. 17-24.

[21]LARSEN, Ronald W. LabVIEW for Engineer. New Jersey: Pearson Education, Inc, 2011

[22]Cotton Incorporated. Estampado Textil. California del Norte : copyright, 2003.

[23] www2.uca.es/grup-invest/ntgc/crealabcp/practicas/scap1.pdf

[24] Geovanny Méndez, Enmanuel Obviedo, Gabriel Fallas, Cristiám Vega, Abel Méndez. Análisis de las herramientas Unity y Blender para el desarrollo de videojuegos con un enfoque educativo. Escuela de computación, tecnologico Costa rica. copyright, 2003.

[25]Festo2016.https://www.festo.com/net/SupportPortal/Files/17269/Ventile_es.pdf

[26]Varitel. Variador DC.<http://www.varitel.com/assets/images/DC-drives.jpg>

ANEXOS

Electroválvulas, válvulas neumáticas, midineumática

FESTO

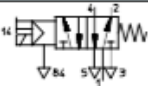
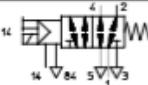
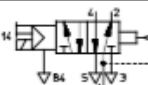
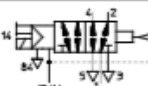


- Válvulas monoestables accionadas eléctrica o neumáticamente
- Válvulas con pilotaje interno o externo
- Montaje en batería
Para 2 ... 10 válvulas
o montaje individual

Electroválvulas MEBH, midineumática

Hoja de datos: válvulas de 5/2 vías

FESTO

| Referencias | | | | |
|---|--|----------|----------------|----------------------------|
| Símbolo | Descripción | Tensión | Nº de artículo | Tipo |
| Recuperación mecánica del muelle, alimentación interna del aire de pilotaje | | | | |
|  | Válvula con conexiones de utilización roscadas | 24 V DC | 173004 | MEBH-5/2-1/8-B |
| | | 110 V AC | 173043 | MEBH-5/2-1/8-B-110AC |
| | Con rosca G1/8 | 230 V AC | 173085 | MEBH-5/2-1/8-B-230AC |
| | Válvula con conexiones de utilización roscadas | 24 V DC | 173006 | MEBH-5/2-1/8-P-B |
| | | 110 V AC | 173057 | MEBH-5/2-1/8-P-B-110AC |
| | para placa de alimentación PRS-ME | 230 V AC | 173099 | MEBH-5/2-1/8-P-B-230AC |
| | Válvula para placa base | 24 V DC | 173005 | MEBH-5/2-5,0-B |
| | | 110 V AC | 173071 | MEBH-5/2-5,0-B-110AC |
| | | 230 V AC | 173113 | MEBH-5/2-5,0-B-230AC |
| Recuperación mecánica del muelle, alimentación externa del aire de pilotaje | | | | |
|  | Válvula con conexiones de utilización roscadas | 24 V DC | 173007 | MEBH-5/2-1/8-S-B |
| | | 110 V AC | 173046 | MEBH-5/2-1/8-S-B-110AC |
| | Con rosca G1/8 | 230 V AC | 173088 | MEBH-5/2-1/8-S-B-230AC |
| | Válvula con conexiones de utilización roscadas | 24 V DC | 173009 | MEBH-5/2-1/8-P-S-B |
| | | 110 V AC | 173060 | MEBH-5/2-1/8-P-S-B-110AC |
| | para placa de alimentación PRS-ME | 230 V AC | 173102 | MEBH-5/2-1/8-P-S-B-230AC |
| | Válvula para placa base | 24 V DC | 173008 | MEBH-5/2-5,0-S-B |
| | | 110 V AC | 173074 | MEBH-5/2-5,0-S-B-110AC |
| | | 230 V AC | 173116 | MEBH-5/2-5,0-S-B-230AC |
| Recuperación neumática del muelle, alimentación interna del aire de pilotaje | | | | |
|  | Válvula con conexiones de utilización roscadas | 24 V DC | 173010 | MEBH-5/2-1/8-L-B |
| | | 110 V AC | 173044 | MEBH-5/2-1/8-L-B-110AC |
| | Con rosca G1/8 | 230 V AC | 173086 | MEBH-5/2-1/8-L-B-230AC |
| | Válvula con conexiones de utilización roscadas | 24 V DC | 173012 | MEBH-5/2-1/8-P-L-B |
| | | 110 V AC | 173058 | MEBH-5/2-1/8-P-L-B-110AC |
| | para placa de alimentación PRS-ME | 230 V AC | 173100 | MEBH-5/2-1/8-P-L-B-230AC |
| | Válvula para placa base | 24 V DC | 173011 | MEBH-5/2-5,0-L-B |
| | | 110 V AC | 173072 | MEBH-5/2-5,0-L-B-110AC |
| | | 230 V AC | 173114 | MEBH-5/2-5,0-L-B-230AC |
| Recuperación neumática del muelle, alimentación externa del aire de pilotaje | | | | |
|  | Válvula con conexiones de utilización roscadas | 24 V DC | 173013 | MEBH-5/2-1/8-L-S-B |
| | | 110 V AC | 173045 | MEBH-5/2-1/8-L-S-B-110AC |
| | Con rosca G1/8 | 230 V AC | 173087 | MEBH-5/2-1/8-L-S-B-230AC |
| | Válvula con conexiones de utilización roscadas | 24 V DC | 173015 | MEBH-5/2-1/8-P-L-S-B |
| | | 110 V AC | 173059 | MEBH-5/2-1/8-P-L-S-B-110AC |
| | para placa de alimentación PRS-ME | 230 V AC | 173101 | MEBH-5/2-1/8-P-L-S-B-230AC |
| | Válvula para placa base | 24 V DC | 173014 | MEBH-5/2-5,0-L-S-B |
| | | 110 V AC | 173073 | MEBH-5/2-5,0-L-S-B-110AC |
| | | 230 V AC | 173115 | MEBH-5/2-5,0-L-S-B-230AC |



Al Servicio de la Automación Industrial

DRIVER DIGITAL DE CORRIENTE CONTINUA

V65D

MANUAL



V5.3 ENERO 08

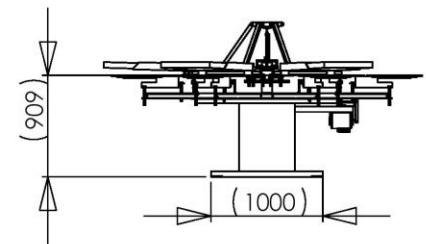
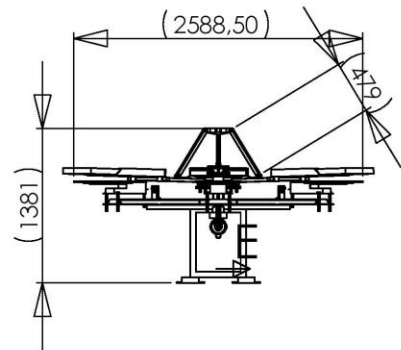
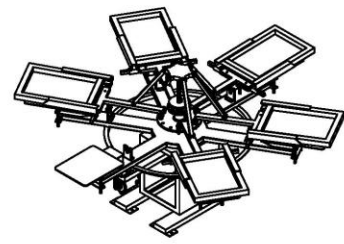
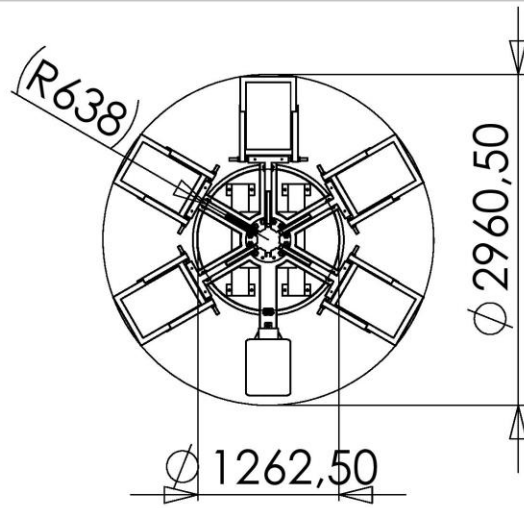
VARITEL INGENIERIA ELECTRONICA SA

Manuel Saifia 485 - Lomas de Zamora (B1632CCJ) B.O.A.S. ARGENTINA - Tel: (54-11) 4243-1171 Fax: (54-11) 4262-7845

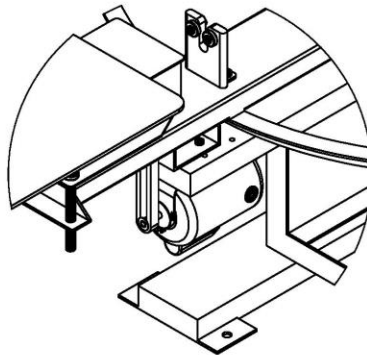
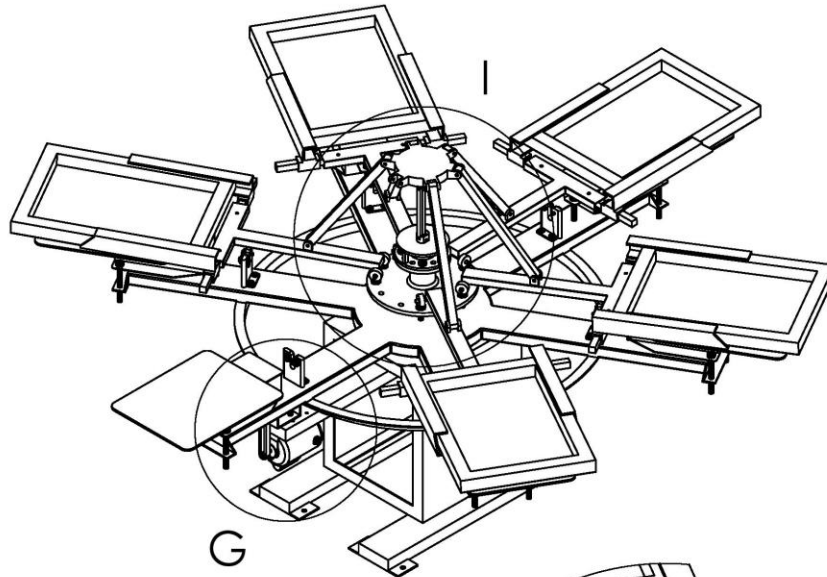
Visite nuestro web site: WWW.VARITEL.COM Email: INFO@VARITEL.COM

1. Specifications

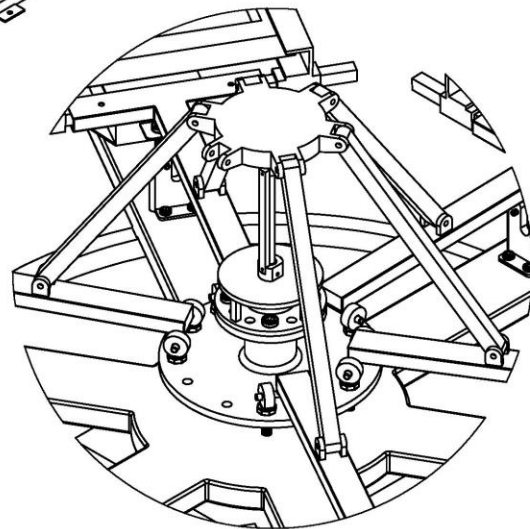
| Item | | Specifications | | | | | | | | |
|--|-----------------------------|---|------|------|--------------------|------|------|------|------|------|
| Type Code | | V6SD-DCD4 | | | | | | | | |
| Type | | 0035 | 0070 | 0110 | 0150 | 0180 | 0270 | 0380 | 0500 | 0720 |
| Maximum Applicable DC Motor (KW) | 220Vdc | 7.5 | 15 | 22 | 30 | 37.5 | 55 | 75 | 90 | 150 |
| | 440Vdc | 15 | 30 | 45 | 60 | 75 | 110 | 150 | 185 | 300 |
| Input Power (VAC) | Main Circuit (□1) | 3□ 200 ~ 440 VAC ±10□ 50/60Hz | | | | | | | | |
| | Control Circuit | 1□ 200/220 VAC ±10□ 50/60Hz | | | | | | | | |
| Rating Output | Armature Current | 35 | 70 | 110 | 150 | 180 | 270 | 380 | 500 | 720 |
| | Field Current (A) | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| Operating Environment | Location | Indoor (avoid with corrosive gas and dust) | | | | | | | | |
| | Ambient Temperature (□2) | 0 ~ +45□ | | | 0 ~ +35□ | | | | | |
| | Humidity | Relative humidity below 85□ (non-condensing) | | | | | | | | |
| | Altitude (□3) | Below altitude 1000M(sea level) | | | | | | | | |
| Structure | Cooling | Self-cooling | | | Forced cooling fan | | | | | |
| | Structure | Wall Mounted, Steel Housing, Protection IP20 | | | | | | | | |
| | Color | Orange | | | | | | | | |
| Characteristic | Accuracy | 2% armature voltage feedback (standard function) | | | | | | | | |
| | | 0.1% DC TC feedback (standard function) | | | | | | | | |
| | | 0.01% Encoder feedback (standard function) | | | | | | | | |
| | Speed Range | 100: 1(DC TG feedback) | | | | | | | | |
| | Overload Ability | 110% continuous; 150%, 30sec; 200%, 10 sec | | | | | | | | |
| | Control Characteristic | Fully Digital Speed Loop: Adjustable P.I Gain Current Loop: Adjustable P.I Gain | | | | | | | | |
| Display Function | Display Function | Digital LCD parameters display and Diagnostic Display | | | | | | | | |
| | | 5 LED Status indication (run, stop,) | | | | | | | | |



| | | | | | | | | | |
|--|--|--|--|---|--|--|--|----------|--|
| SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM TOLERANCIAS: ± 0.5 | | | | PROPIEDAD INTELECTUAL Y CONFIDENCIALIDAD LA INFORMACIÓN CONTENIDA EN ESTE DIBUJO ES PROPIEDAD EXCLUSIVAMENTE DE CFOUR S.A.S. CUALQUIER REPRODUCCIÓN PARCIAL O TOTAL SIN AUTORIZACIÓN ESCRITA POR CFOUR S.A.S. ESTÁ PROHIBIDA. | | NO CAMBIE LA ESCALA | | REVISIÓN | |
| | | | | | | CFOUR S.A.S. | | | |
| | | | | | | TÍTULO: DISEÑADO POR: BRIAN CHACON HERNANDEZ | | | |
| | | | | | | ENSAMBLAJE TOTAL | | | |
| | | | | | | A4 | | | |
| DIBUJ. VERIF. APROB. FABR. CALID. | | | | MATERIAL: ACERO INOXIDABLE, MADERA | | ESCALA: 1:50 | | | |
| | | | | PESO: | | HOJA 1 DE 4 | | | |

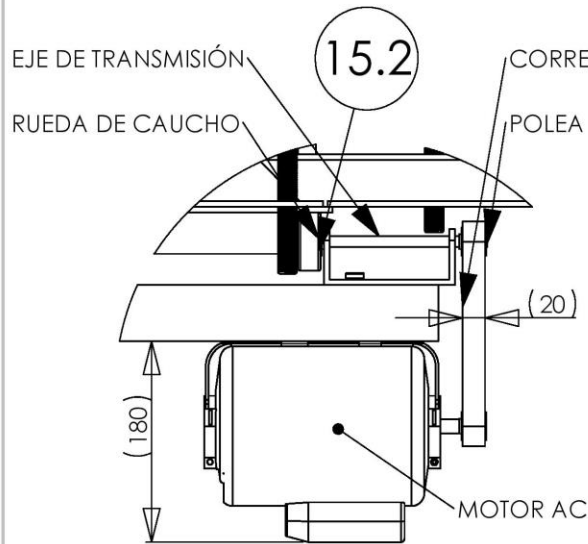
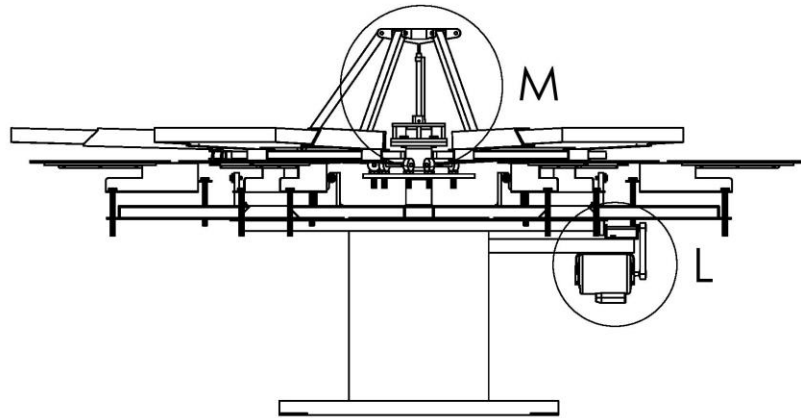


DETALLE G
ESCALA 1 : 10

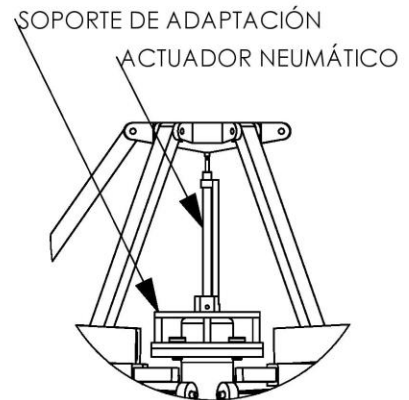


DETALLE I
ESCALA 1 : 10

| | | | | | | | | | |
|--|--|--|--|---|--|--|--|-------------|----|
| SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM TOLERANCIAS: ± 0.5 | | | | PROPIEDAD INTELECTUAL Y CONFIDENCIALIDAD LA INFORMACIÓN CONTENIDA EN ESTE DIBUJO ES PROPIEDAD EXCLUSIVAMENTE DE CFOUR S.A.S. CUALQUIER REPRODUCCIÓN PARCIAL O TOTAL SIN AUTORIZACIÓN ESCRITA POR CFOUR S.A.S. ESTÁ PROHIBIDA. | | NO CAMBIE LA ESCALA | | REVISIÓN | |
| | | | | | | CFOUR S.A.S. | | | |
| DIBUJ. VERIF. APROB. FABR. CALID. | | | | | | TÍTULO: DISEÑADO POR: BRIAN CHACON HERNANDEZ | | | |
| MATERIAL: ACERO INOXIDABLE, MADERA | | | | | | ENSAMBLAJE TOTAL | | | A4 |
| PESO: | | | | | | ESCALA: 1:20 | | HOJA 2 DE 4 | |

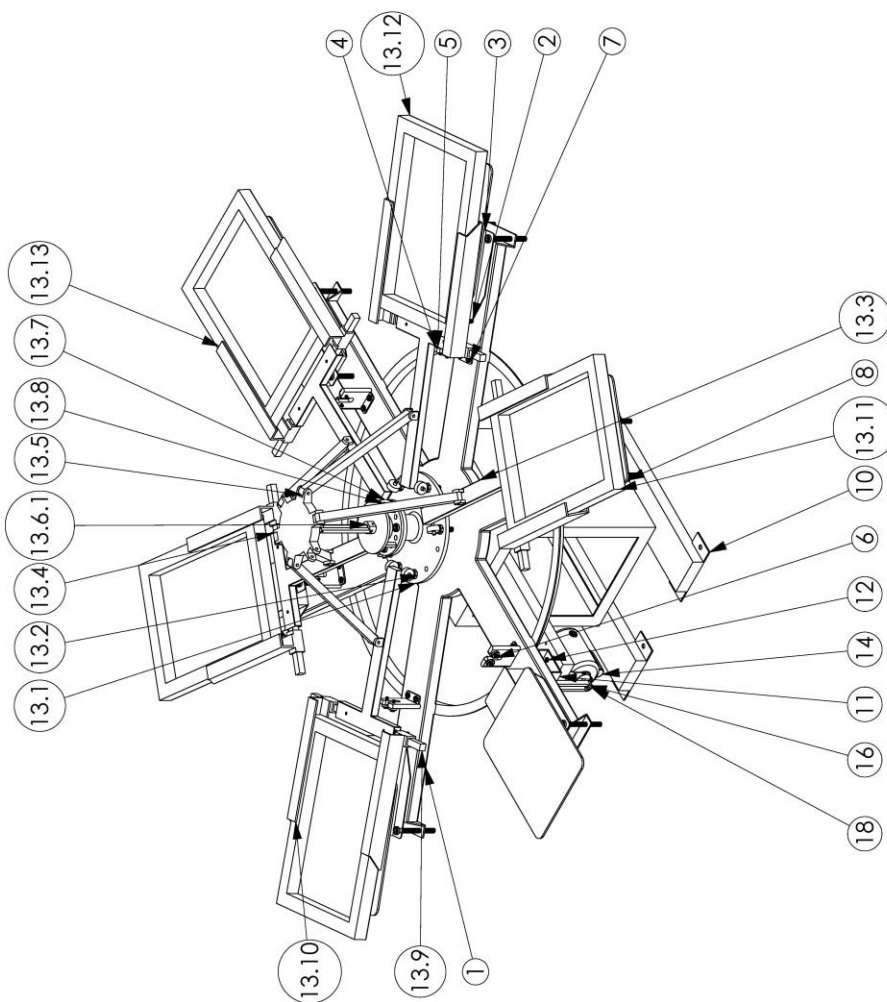


DETALLE L
ESCALA 1 : 5



DETALLE M
ESCALA 1 : 10

| | | | | | | | | | |
|--|--|--|--|---|--|---|--|----------|--|
| SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM TOLERANCIAS: ± 0.5 | | | | PROPIEDAD INTELECTUAL Y CONFIDENCIALIDAD LA INFORMACIÓN CONTENIDA EN ESTE DIBUJO ES PROPIEDAD EXCLUSIVAMENTE DE CFOUR S.A.S. CUALQUIER REPRODUCCIÓN PARCIAL O TOTAL SIN AUTORIZACIÓN ESCRITA POR CFOUR S.A.S. ESTÁ PROHIBIDA. | | NO CAMBIE LA ESCALA | | REVISIÓN | |
| | | | | | | CFOUR S.A.S. | | | |
| | | | | TÍTULO: | | DISEÑADO POR: BRIAN CHACON HERNANDEZ | | | |
| DIBUJ. VERIF. APROB. FABR. CALID. | | | | MATERIAL: ACERO INOXIDABLE, MADERA | | ENSAMBLAJE TOTAL | | | |
| | | | | | | A4 | | | |
| | | | | ESCALA: 1:20 | | HOJA 3 DE 4 | | | |

[illegible][illegible]

Codigo C# recepción, modelos y movientos de la máquina

```
using UnityEngine;
using System.Collections;

using System;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.Threading;
using System.Text.RegularExpressions;

using System.Linq;

using System.Collections.Generic;

public class prueba : MonoBehaviour {

    // receiving Thread
    Thread receiveThread;

    // udpclient object
    UdpClient client;

    // public
    // public string IP = "127.0.0.1"; default local
    public int port; // define > init

    // infos
    public string lastReceivedUDPPacket="";
    public string allReceivedUDPPackets=""; // clean up this from time to time!
    public string entrada;
    public string DF;
    public int DA;
    public int DB;
    public static int DC;
    public int DE;
    public int i=0;
    public int result;
    public int dd;
    public float total=0;
    public float salto=0.03f;
    Transform __cuandrado;
    Transform __cilindro;
```

```

private static void Main()
{
    prueba receiveObj=new prueba();
    receiveObj.init();

    string text="";
    do
    {
        text = Console.ReadLine();
    }
    while(!text.Equals("exit"));
}
public void Start()
{
    init();
}
public void init()
{
    print("UDPSend.init()");
    port = 8051;

    print("Sending to 127.0.0.1 : "+port);
    print("Test-Sending to this Port: nc -u 127.0.0.1 "+port+"");

    receiveThread = new Thread(
        new ThreadStart(ReceiveData));
    receiveThread.IsBackground = true;
    receiveThread.Start();
}

public void ReceiveData()
{
    client = new UdpClient(port);
    while (true)
    {
        try
        {

```

```

        IPEndPoint anyIP = new IPEndPoint(IPAddress.Any, 0);
        byte[] data = client.Receive(ref anyIP);
        string text = Encoding.UTF8.GetString(data);
        print(">> " + text);

    entrada = lastReceivedUDPPacket;
    string pattern = "("; // Split on hyphens

    string[] ja = Regex.Split(entrada, pattern);
    i=0;
    foreach (string value in ja)
    {

        if(i==1)
            DA=result;
        if(i==2)
            DB=result;
        if(i==3)
            DC=result;
        if(i==4)
            DE=result;

        try {
            result = Convert.ToInt32(value);
            Console.WriteLine("Converted the {0} value '{1}' to
the {2} value {3}.",
                                value.GetType().Name, value,
                                result.GetType().Name, result);
            i++;
        }

        catch (OverflowException) {
            Console.WriteLine("{0} is outside the range of the
Int32 type.", value);
        }
        catch (FormatException) {
            Console.WriteLine("The {0} value '{1}' is not in a
recognizable format.",
                                value.GetType().Name, value);
        }
    }

    lastReceivedUDPPacket=text
    allReceivedUDPPackets=allReceivedUDPPackets+text;
}

```



```

        catch (Exception err)
        {
            print(err.ToString());
        }
    }
}

public void Update() {

    var cuadrado=GameObject.FindGameObjectWithTag("cuca");
    var cilindro=GameObject.FindGameObjectWithTag("chimbo");

    __cuadrado = cuadrado.transform;
    __cilindro = cilindro.transform;

    if (DA == 24) {
        total += DC ;
        aquitambien();
    }

    else if (DC == 24) {
        total += DC ;
        aquifue();
    } else if (DC == 0) {
        __cilindro.localPosition=new Vector3(0,1.45f,0);//(0,0.49f,0);//(0, -total, 0);
    }

    public void aquifue() {

        public float Varma=0.2*DC*cilindro.localPosition;
        public float Varmb=-0.1*DC* last.cilindro.localPosition;
        public float Varmc=0.996* DC*last.cilindro.localPosition;
        public float Varmabc = Varma+ Varmb+ Varmc;
        __cilindro.localPosition=new Vector3(0,Varmabc,0);

    }

    public void aquitambien() {
        public float vare = 0.0080408*DA;
        public float varf = 0.997* last.cuadrado.localPosition;
        public float varef = vare*varf;
        __cuadrado.transform.Rotate(0,0, varef * Time.deltaTime);
    }

    public string getLatestUDPPacket()
    {
        allReceivedUDPPackets="";
    }
}

```

```

        return lastReceivedUDPPacket;
    }
    void OnGUI()
    {
        Rect rectObj=new Rect(40,10,200,400);
        GUIStyle style = new GUIStyle();
        style.alignment = TextAnchor.UpperLeft;
        GUI.Box(rectObj,"# UDPReceive\n127.0.0.1 "+port+" #\n"
            + "shell> nc -u 127.0.0.1 : "+port+" \n"
            + "\nultimo mensaje: \n"+ lastReceivedUDPPacket
            + "\n\n todos los mensajes: \n"+allReceivedUDPPackets
            + "\n\n numero Y0: \n"+ DA
            + "\n\n numero Y1: \n"+ DB
            + "\n\n numero X0: \n"+ DC
            + "\n\n numero X1: \n"+ DE
            ,style);
    }
}

```

Codigo C# envió de posición de los elementos virtuales a labview

```
using UnityEngine;
using System.Collections;

using System;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.Threading;

public class UDPSend : MonoBehaviour
{
    public static int localPort;

    public string IP;
    public int port;
    public string perro;
    public float pez;
    IPEndPoint remoteEndPoint;
    UdpClient client;

    public string strMessage="";

    public static void Main()
    {
        UDPSend sendObj=new UDPSend();
        sendObj.init();

        sendObj.inputFromConsole();

        sendObj.sendEndless(" endless infos \n");

    }
    public void Start()
    {
        init();

    }

    // OnGUI
    void OnGUI()
    {
        Rect rectObj=new Rect(40,380,200,400);
        GUIStyle style = new GUIStyle();
```

```

        style.alignment = TextAnchor.UpperRight;
        GUI.Box(rectObj, "# UDPSend-Data\n127.0.0.1 "+port+"#\n"
            + "shell> nc -lu 127.0.0.1 "+port+" \n"
            ,style);

        sendString(strMessage+"\n");

    }
    public void init()
    {
        print("UDPSend.init()");

        IP="127.0.0.2";
        port=8060;
        remoteEndPoint = new IPEndPoint(IPAddress.Parse(IP), port);
        client = new UdpClient();
        var cuadrado = GameObject.FindGameObjectWithTag("cuca");

        var cilindro = GameObject.FindGameObjectWithTag("chimbo");

        float poprueba1 = cuadrado.transform.rotation.y;
        float poz=cuadrado.transform.rotation.z;
        float pox=cuadrado.transform.rotation.x;
        float poy=cuadrado.transform.rotation.y;
        pez = poprueba1;

        float poprueba2 = cilindro.Local.position.y;
        float poz=cilindro.Local.position.z;
        float pox=cilindro.Local.position.x;
        float poy=cilindro.Local.position.y;
        pezito = poprueba2;

        print("Sending to "+pez+" "+pezito+" "+IP+" : "+port);
        byte[] data = Encoding.UTF8.GetBytes(pez.ToString());
        client.Send(data, data.Length, remoteEndPoint);
    }

    public void inputFromConsole()
    {
        try
        {
            string text;
            do
            {

```

```

        text = Console.ReadLine();

        if (text != "")
        {
            byte[] data = Encoding.UTF8.GetBytes(text);
            client.Send(data, data.Length, remoteEndPoint);
        }
    } while (text != "");
}
catch (Exception err)
{
    print(err.ToString());
}
}

public void sendString(string message)
{
    try

        var cuadrado = GameObject.FindGameObjectWithTag("cuca");
        var cilindro = GameObject.FindGameObjectWithTag("chimbo");

        Vector3 pos = cuadrado.transform.position;
        pos.x = Mathf.RoundToInt(gameObject.transform.position.x);
        pos.y = Mathf.RoundToInt(gameObject.transform.position.y);
        pos.z = Mathf.RoundToInt(gameObject.transform.position.z);
        transform.position = pos;

    }
    catch (Exception err)
    {
        print(err.ToString());
    }
}

public void sendEndless(string testStr)
{
    do
    {
        sendString(testStr);
    }
    while(true);
}
}

```

Identificación del modelo de un motor

PRÁCTICA 1: "Identificación del modelo de un motor de C.C. con entrada en escalón de tensión"

Objetivos:

- Caracterizar un motor de C.C.
- Determinar las constantes K y τ .
- Simulación del funcionamiento de un motor de C.C. en ViSim.

Guión:

- Principios de funcionamiento de un motor de C.C. de excitación independiente.
- Conocimientos teóricos del modelo matemático del motor de C.C. de excitación independiente.
- Ensayo experimental.
- Simulación.

-1- Motor de C.C. de excitación independiente

-2- Función de transferencia de un motor de C.C.

Un motor de c.c. está formado por un estator o inductor que es la parte fija del motor y un rotor o inducido que es la parte móvil. El desarrollo de los elementos, para crear el modelo matemático, tanto del rotor como del estator, una vez que se les somete a una d.d.p., es el que muestra la figura siguiente.

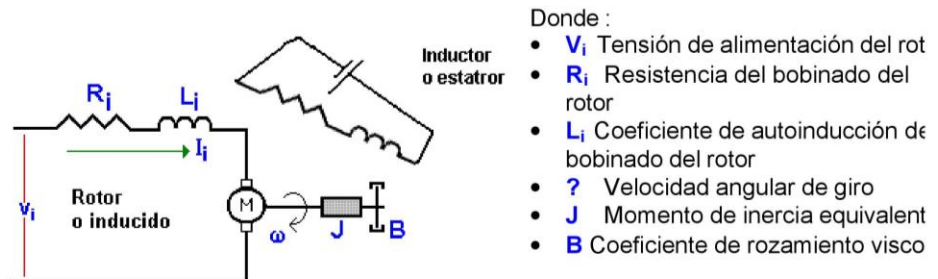


Fig. 1. Esquema de un motor de C.C.

En la figura 1, podemos observar que no existe conexión eléctrica entre el rotor y el estator.

Determinamos ahora el diagrama de bloques de nuestro objeto motor.



Donde:

- V_i Tensión de alimentación del rotor
- ω Velocidad angular
- $G(s)$ Función de transferencia del motor
 - K Ganancia del sistema
 - s Variable compleja
 - τ Constante de tiempo

Fig. 2. Función de transferencia de un motor de C.C.

Determinar el modelo matemático

1º) Que sucede en el inducido

- Al aplicar una tensión V_i al inducido, circula por el una corriente I_i .
- Por circular esta corriente I_i , por el rotor, se inducirá una fuerza contra electromotriz (ley de Lenz " *toda corriente se opone a la causa que la produce* ") cuyo valor vendrá determinado por la expresión

$$V_b = K_b \times \omega \quad (1)$$

siendo K_b la constante de fuerza contraelectromotriz y ω la velocidad angular.

Aplicando la ley de Ohm la tensión útil será:

$$V_i(t) - K_b \times \omega(t) = R_i \times i(t) + L_i \frac{di_i(t)}{dt} \quad (2)$$

Considerando $L_i \cong 0$ H, entonces:

$$V_i(t) - K_b \times \omega(t) = R_i \times I_i(t) \quad (3)$$

El motor en su movimiento giratorio arrastra una carga, creándose por lo tanto, un par motor. Este será pues igual a:

$$K_p \times I_i(t) = J \times \frac{d\omega(t)}{dt} + B \times \omega(t) \quad (4)$$

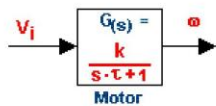
Suponiendo $B \approx 0$

$$K_p \times I_i(t) = J \frac{d\omega(t)}{dt} \quad (5)$$

Despejando $I_i(t)$ de la segunda ecuación y substituyéndola en la primera. Nos quedará una ecuación diferencial de primer orden (solo aparece la primera derivada), de grado 1 (la incógnita ω está elevada a 1) y con coeficientes constantes.

$$\frac{R_i \times J}{K_p} \times \frac{d\omega(t)}{dt} + K_b \times \omega(t) = V_i(t) \quad (6)$$

El procedimiento de resolución de la ecuación diferencial, se puede realizar mediante el empleo de la transformada de Laplace (anexo 1) o mediante integración directa.



$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K}{s \tau + 1} \quad (7)$$

$$K = \frac{1}{K_b} \quad (8)$$

$$\tau = \frac{R_i \times J}{K_p \times K_b} \quad (9)$$

Ensayo experimental

- Introducimos una tensión de entrada en escalón del tipo $t \leq 0 = 0V$, $t > 0 = V_i$

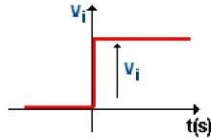


Fig. Oscilograma de la señal de entrada

Aplicando este tipo de señal obtendremos de la ecuación diferencial:

$$\omega(t) = K \times V_i \times \left[1 - e^{-\frac{t}{\tau}} \right] \quad (10)$$

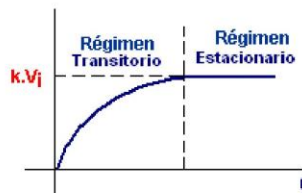
¿Cuál será la velocidad angular cuando t tiende a infinito?

Ya que el cociente $-\frac{t}{\tau}$ tiende a 0 el
 $\lim_{t \rightarrow \infty}$ es
 igual a $K.V_i$

$$\lim_{t \rightarrow \infty} K \times V_i \left[1 - e^{-\frac{t}{\tau}} \right] = K.V_i \quad (11)$$

Mientras que en el instante inicial

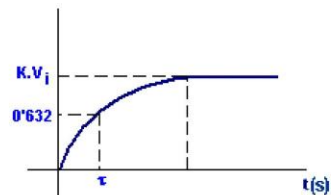
$$\lim_{t \rightarrow 0} \omega(t) = 0 \quad (12)$$

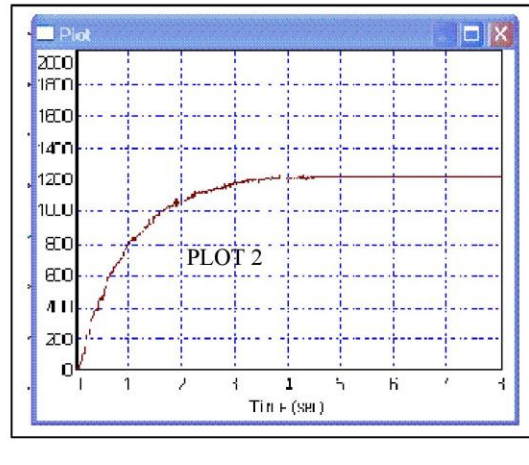


τ Es el tiempo que tarda el sistema en alcanzar el 63,2 % del valor final. A partir de τ 0

si hacemos $t = \tau$ entonces

$$\omega(t=\tau) = K \times V_i \left[1 - \underbrace{\frac{1}{e}}_{=0.632} \right] = 0.632 \times K \times V_i$$





Del plot 2 se observa que el valor estacionario de la velocidad es:
1219'04 r.p.m.

Determinaremos ahora **K**

$$K \times V_i = ? = 1219'04 \Rightarrow K = \frac{?}{V_i} = \frac{1219'04}{4'357} = 279.7888 \frac{\text{rpm}}{\text{V}}$$

Para determinar ?

Anexo 1

Resolución de la ecuación diferencial por la transformada de Laplace

La aplicación de la transformada de Laplace en ambos miembros de la ecuación 6 conduce a que:

$$\left(\frac{R_i \times J}{K_p} \times s + K_b \right) \times \omega(s) = V_i(s) \quad (a1)$$

Siempre que se suponga la velocidad del motor nula en el instante inicial, es decir $\omega(t=0)=0 \text{ r.p.m.}$.

Reordenando (a1) se obtiene la función de transferencia del motor.

$$\boxed{\frac{\omega(s)}{V_i(s)} = \frac{1/K_b}{\frac{Ri \times J}{K_p \times K_b} \times s + 1} = \frac{K}{s \times \tau + 1}} \quad (\text{a2})$$

$$\text{Siendo } K = \frac{1}{K_b} \text{ y } \tau = \frac{K_i \times J}{K_p \times K_b}$$

Aplicación de la transformada de laplace en (10)

Si aplicamos al motor un escalón de tensión de valor V_i , su transformada de laplace será:

$$V_i(s) = \frac{V_i}{s} \quad (\text{a3})$$

De forma que sustituyendo en (7) se obtiene

$$\omega(s) = V_i \times K \times \frac{1}{s \times (s \times \tau + 1)} \quad (\text{a4})$$

Expresión que puede ponerse en la forma:

$$\omega(s) = V_i \times K \times \left[\frac{1}{s} - \frac{\tau}{s \times \tau + 1} \right] = V_i \times K \times \left[\frac{1}{s} - \frac{\tau}{s + 1/\tau} \right] \quad (\text{a5})$$

Aplicando la transformada inversa de Laplace.

$$\boxed{\omega(t) = L^{-1}[\omega(s)] = V_i \times K \times \left[1 - e^{-\frac{1}{\tau} \times t} \right]} \quad (\text{a6})$$

Resolución de la ecuación diferencial por integración directa

La ecuación diferencial se puede poner en la forma:

$$\boxed{\frac{d\omega(t)}{dt} + R \times \omega(t) = Q} \quad (\text{a7})$$

$$\text{Siendo } R = \frac{K_b \times K_p}{R_i \times J} \text{ y } Q = \frac{V_i \times K_p}{R_i \times J}$$

Reconsidera que:

$$\frac{d}{dt} \times [\omega(t) \times e^{R \times t}] = \frac{d\omega(t)}{dt} \times e^{R \times t} + \omega(t) \times R \times e^{R \times t} \quad (\text{a8})$$

Multiplicando miembro a miembro por $e^{-R \times t}$

$$\boxed{e^{-R \times t} \times \frac{d}{dt} \times [\omega(t) \times e^{R \times t}] = \frac{d\omega(t)}{dt} + \omega(t) \times R} \quad (\text{a9})$$

Sustituyendo (a9) en (a7)

$$\frac{d}{dt} \times [\omega(t) \times e^{R \times t}] = Q \times e^{R \times t} \quad (\text{a10})$$

$$\omega(t) \times e^{R \times t} = Q \times \int e^{R \times t} \times dt \quad (\text{a11})$$

$$\boxed{\omega(t) \times e^{R \times t} = \frac{Q}{R} \times e^{R \times t} + Cte} \quad (\text{a12})$$

La constante de integración se calcula a partir de las condiciones iniciales.

$$t = 0, \omega = 0, \text{ y a partir de (a12)} \rightarrow C = -\frac{Q}{R}$$

De forma que sustituyendo en (a12)

$$\omega(t) \times e^{R \times t} = \frac{Q}{R} e^{R \times t} - \frac{Q}{R} \quad (\text{a13})$$

$$\boxed{\omega(t) = \frac{Q}{R} \times [1 - e^{-R \times t}]} \quad (\text{a14})$$

Considerando los valores de R y Q que haga a (10)